SINCLAIR QL WORLD

Volume II: Issue 5: Price £2.50

Standard Deviation

QDS

QMATHS

Algebraic Processor

Transfer Utility

JDIR

ISSN 0951-9335

SINCLAIR QL WORLD

VOL 2 ISSUE 5

Contents

Editor
Helen Armstrong

Publisher
Mark Kasprowicz

Advertising Manager
Jim Peskett

Creative Director
John Stanley

Graphic Artist
Stevie Billington

Magazine Services
Linda Miller, Frances Maxwell, Pauline Wakeling

Sinclair QL World,
Published by Arcwind Ltd.
The Blue Barn,
Tew Lane, Wootton,
Woodstock,
Oxon. OX7 1HA
Tel: 0993 811181
Fax: 0993 811481
ISSN 026806X

If you have any comments or difficulties please write to the editor and we will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2.50 UK, £2.99 Europe. Overseas rates on request.

Subscriptions from: Arcwind The Blue Barn, Tew Lane, Wootton, Woodstock, Oxon. OX7 1HA
UK: £23.40
Europe: £32.90
Rest of World: £40.90

Reprographic Services: Eclipse, Brook Street, Watlington, Oxon. OX9 5JH.
Distributed by: Seymour Press Ltd., Windsor House, 1270 London Road, Norbury, London, SW16 4DH

© 1993 ARCWIND LTD.
Sinclair QL World is published 12 times a year. All rights reserved. Reproduction in whole or in part without written permission is strictly prohibited. We welcome contributions. All material must be supplied with a S.A.E.

While all reasonable care is taken in compiling QL World, the publisher and its agents assume no responsibility in affects arising therefrom. Opinions expressed are those of the authors.

8 TROUBLESHOOTER
Bryan Davies unhogs memory, sizes screens and swaps roms.

11 QL SCENE
Software87 into DTP ... QXL heads for market ... ILQR USA show ... games from Merz ... New DIY Toolkit agent... Postie returns.

12 OPEN CHANNEL
Sideways printing ... colour printing ... machine code queries ... EOL puzzle ... SuperBasic error trapping ... hardware resisting.

20 VERY BASIC SUPERBASIC
Dilwyn Jones embarks on some practical excercises in Part 2 of his tutorial.

23 ARCHIVE ANSWERS - STANDARD DEVIATION
An average hobby for Robin Stevenson - using mean statistics!

27 DIY TOOLKIT
Simon Goodwin's HP_DUMP screen image printer for page printers.

30 REVIEW - TEXT'n'GRAPHIX
Henry Orlowski tries a low-cost program to combine WP text and imported graphics.

32 REVIEW: QMATHS 2.00
We turn Hilary Snaden loose on the latest gener ation of DP's maths package.

38 SUPERBASIC IN ACTION
Simon Goodwin makes last month's PAGE_PRINT more versatile.

40 REVIEW: TRANSFER UTILITY
Bryan Davies tries the latest file-handling update.

42 INSTANT ACCESS
At-a-glance guide to your software and hardware suppliers.

43 REVIEW: JDIR
At last! A low-cost easy means of controlling your DIR, says Bryan Davies.

44 REVIEW: QL-HDD-CARD
Taking the hard disk plunge with a Falkenburg system, by Peta Jaeger.

47 MICRO ADS

Coming Soon

Alan Bridewell should be back with Beginners Machine code, and we have part four of Henry Orlowski's Easy With Easel. Rich Mellor is building his system up with Jurgen Falkenburg's hard disk interface and a special custom box from David Johnston. And we'll be looking for Mike Lloyd. . .

TROUBLE SHOOTER

Memory hogs, screen size and rom swapping with Bryan Davies.

Time to clear out a few items that have been sitting, awaiting attention, for some time. My mention, in the January issue, of the way Quill (and other Psion programs) hog memory prompted a response from Ray Dent, who sent in the code he uses for handling these programs. This is the Taskforce routine, from DIY Toolkit in April 1989, and is available as part of Simon Goodwin's DIY Tookit Disk J (see QL Scene for current ordering details). (Please note that the DIY Toolkit disks are NOT Public Domain. CGH Services, handled them happily until it left the QL market, had a long list of copyright programs, which are still copyright, usually to their original authors Editor.)

This routine is clearly rather more than a method of restricting the space taken by Quill. Dent remarks that "It seems to slow up a bit if you have more than five or so programs running". That should not worry most users much.

Ramdisks

Richard Kettlewell wrote from Cambridge University on the same subject. Firstly, he suggested formatting a fixed-size ramdisk, to keep Quill from stealing that amount of space, but my experience trying that was negative, in so far as subsequent attempts to get the space back (on a Trump Card-equipped system) failed because all free space is put into RAM disk(s) and attempting to reduce the size of one simply puts the space into another. This is because the type of disk is dynamic; there is no problem with the static type. As an alternative, he suggested dimensioning a large array initially, starting Quill, then going back to SuperBasic and using CLEAR to free the array space up. You would need to have modified Quill so that it works from the EXEC command, rather than EXEC_W, to be able to do this.

Talking about arrays is surely going above the heads of most users, though. As with the ramdisk approach, it is perhaps unsuitable simply because more knowledge is required, to achieve the desired result, than the typical user will have. Richard mentioned the use of the *QRam/QPac* HOT_REXT and EXEP as another way of dealing with Quill, and suggested they will become "increasingly necessary for the most up-to-date software". To some extent, I agree; undoubtedly, there has been more mention of QPac in my (personal) mail recently, and the trumpeting of the Windows GUI on PCs may have a rub-off effect on the QL scene, but the problem with QPac is that the cure tends to be worse than the ailment. It is - to put it very mildly - not easy to understand how to set it up.

Quill users do not strike me as wanting to plough through pages written in some weird language; they are likely to want something straightforward, and written in plain English.

Exclusive!

Richard had some other interesting comments to make. One was that the main storage of the university computer seems to work with 42 KB sectors. As he says "efficient or what?!". We think in terms of 512-byte sectors, or maybe even 1 KB sectors, but what can you store if the minimum unit is 42 KB? He wonders about some things that never seem to make it into QL World, such as Tony Tebby's SMS-2 operating system, as used (I think) on the Atari QL emulator, and file compression/decompression programs (eg Arc). We would print information on such matters, Richard, if the writers/suppliers gave us something above the level of hints. *(I spoke with someone who was working with a forerunner of SMS-2 some time back. He asked me not to print anything because they didn't know how it was going to be applied. That was two years ago. SMS2 itself has not impinged on mainstream QL use until this very month, when it appears on the scene in its latest incarnation, the appropriately-named SMSQ, as the operating system for the Miracle Systems QXL board. Ed.)* They either are not interested in publicity or perhaps feel public magazines "aren't their scene", which may sound rather strange to readers QL World keeps space for program users, as opposed to those whose main interest is programming itself, but program writers seem to be interested only in other programmers. This attitude has, of course, been a major influence in the relative lack of commercial success of the QL (compare it with the Amstrad PCW, which is very much a users' computer, of which both the early as well as current models are still very popular). To complete his letting-off of steam, Richard commented on the ANSI-compliant status of C68, which he thinks is far from compliant "It is a K&R compiler with a few ANSI-like extensions, the most important of which share ANSI syntax but have totally different semantics." Well, it's the first time I've heard that point of view.

Screen width

On a separate subject, Ray Dent asked if it is common for the cursor and the first character (on the command line) to be lost off the left of the screen, with a Microvitec Cub display. As the same thing has been happening on my Cub recently, the question prompted me to look back on notes about the Cub, to see where the adjustment controls are. On the main PCB inside the display unit, roughly a third of the way down the left side as viewed from the front, there is a screw adjustment marked Width, which should allow the lost characters to be brought back onto the screen. In the middle at the rear of the PCB, there is a control marked Height, which perhaps needs adjusting too, to balance what is done to the width. Always observe the usual precautions when working inside a display unit, On or Off - THERE IS HIGH VOLTAGE in there during operation, and it takes some time for the voltage to leak away after switching off, so let it stand then for at least half an hour before opening the case up.

The Cub has been a very stable display, on my system. The only occasions that I have felt impelled to adjust it have been when characters started looking slightly multi-coloured, and there are adjusters for White, Green, Blue and Red on the small PCB attached to the end of the display tube which can be used to get rid of this effect.

A disk oddity that is new to me was reported by Thomas Robbins of Quanta. He has HD drives connected to a Gold Card and finds that they routinely format DD disks to HD capacity. He did not say whether or not the full HD capacity is obtained, but he noted that such disks cause compatibility problems in the drives of other QL users. To prevent the HD format, he uses a command that is not listed in the Gold Card instructions - FLP_DENSITY D. This forces a DD format. Having tried this command using D, H and E, I can confirm that it is accepted without murmur. It has no apparent effect when DD disks are formatted in ED drives, the resulting capacity being 1440 sectors for all three parameters. With HD and ED disks, the "format failed" message comes up if attempts are made to format them to less than their nominal capacity, but proper formats are obtained when the parameter used corresponds to the nominal capacity.

It appears that using this command as a way of formatting disks to less than their nominal capacity will be unsuccessful, but use of the appropriate parameter does ensure that disks are formatted to their correct capacity. However, in ED drives, it seems best to either not use the command at all, or use the E parameter; doing the latter results in correct formats on the four disk types checked - standard DD, HD and ED, and DD with a "HD hole" drilled in it. There may well be a sensing difference between Miracle HD and ED drives, as the report from Robbins suggests HD drives do not sense disk type by looking for the square cutouts, whereas the ED drives do look for them. As you might expect, there is a small reservation concerning using flp_density E; drilled DD disks consistently gave about 30 sectors less when E was used than when H was used.

Rom swapping

Many users will have upgraded their QLs from the AH rom to the JM version and, so far as I knew until recently, that could be done simply by removing the existing two chips and replacing them by the newer ones. That procedure worked fine with one of my QLs. Ian Turner wrote recently to the Quanta newsletter to say he had tried this modification and it did not work on his QL; he asked for help. He soon got replies, and the answer to the problem. Earlier QLs had Qdos on eprom chips, while later ones used rom chips. Those with the eproms had two jumpers - JU1 and JU5 - fitted; those with roms had three jumpers fitted - JU2, JU3 and JU4. The jumpers are located alongside IC34 on the PCB and, as Ian commented, they look rather like resistors.

Does anyone out there have a copy of ICE? Ian's copy has become corrupted; it appears to have been the pre-mouse, pre-rom variety (did such a thing exist?).

QXL pricing

Information given out at a recent meeting in Nottingham set the price range for the Miracle QL-emulator card as roughly £300-500, including shipping and VAT. The basic card would have 1 MB of ram, and the highest-priced one 8 MB. This looks to be, prospectively, the fastest "QL" yet, with the clock rate starting at 20 MHz and capable of being lifted to over 30 MHz. In other respects it will have much the effect of the Gold Card. (See QL Scene - Ed.)

Dilwyn Jones Computing is now selling the latest vector drawing program from Progs of Belgium. While this is rather expensive, it does come with masses of clip art and it takes drawing into another league by avoiding much of the jaggedness inherent in the conventional methods. Founts do not become rougher as they are enlarged; in fact, it is reported that those included with this program are actually smoother above about the 10-point size. Software87 also is selling the program, but the approach is different. This supplier offers it with an integrating module to allow text output from text87 Plus4 to be combined with images from the drawing program and printed together from the latter. This does not quite make Plus4 into a pukka DTP program, but many users want only to make straightforward insertions of images into mainly-text documents and the combination of Plus4, the drawing program, and the integration module provides this capability.

Some features of programs influence the user more than others, and so it is with me in the matter of reformatting text in a wordprocessor. The incorporation of automatic reformatting into Perfection improves the feel of the program greatly. On a system with Gold Card, the default auto-reformat setting works fine, the text snapping to its new position very smartly after a pause of 1.5 seconds, without causing significant visual disturbance. The user is provided with control over the time taken for reformatting to occur following modification of text, and this should allow adequately for differences in speed of QL system versions. With the GC fitted, the re-drawing of the screen is fast and the delay can be brought down to almost zero without causing serious lags, but low values do introduce text flicker (even when there is no reformatting) and it seems best to stick with the default setting, or maybe a little below it. If you are the kind of typist that has to look at the keyboard most of the time, you aren't going to notice much of what happens on the screen, and a low-delay setting should be fine. Cursor movement also is smooth and rapid, with the cursor staying clearly visible all the time. My version of Perfection is 5.03 but that may well have been superseded by now.

Cheap bits

Greenweld Electronics (see Information) regularly offer items which will be of interest to the DIY enthusiast. Their February and Spring catalogues list the ICL OPD circuit board complete with 68008 chip for £2.50, and 5.25-inch 360-KB disk drives for £12.95 (£7.00 if you fancy buying 100 or more!). They also have a 12-inch amber monitor for £34.95, complete with internal audio amplifier and speaker; whether or not this could be used with the QL or other micro is not stated, but the input required is composite video with negative synch, not RGB.

The war game program Fleet Tactical Command was released late in 1992 in its FTCII version for PCs, making it possible to link them to QLs for joint operations. Di-Ren (see Information) gave the QL-PC serial wiring connections shown in the illustration here; as they state, they (and QL World) cannot guarantee this wiring information to be correct and, bearing in mind there is voltage on some lines, errors could cause damage, so it is up to the reader to check that the wiring is appropriate for his/her computers.

Laserprinting

Paul Stewart wrote many months ago, and (hopefully) he will have found an answer to his question by now, but the subject is worth mentioning again. He uses *Perfection*, and wanted to use his Panasonic KXP-4430 laser with it. Knowing that a Canon LBP-4 worked satisfactorily when the Brother HR15 Quill printer-driver was adapted for Perfection, he felt he would be successful with the Panasonic. Are you with me so far? He tried using the adapted Brother driver with his printer. Not surprisingly (for those familiar with laser printers), he didn't get sensible results. He then resorted to using an Epson driver from Quill, with the printer in Epson-emulation mode. This got him print, but not in Times as he wanted. Times appears to be available only with the printer's Hewlett-Packard emulation mode. Digital Precision advised using Professional Publisher (which he has got) as there is a Hewlett-Packard LaserJet driver for that, but he did not have time to learn to use ProPub more effectively.

The alternative was to write a driver for the Hewlett-Packard mode of the printer, in Perfection, which allows you to write more-or-less what you want into a driver. The trouble is that the HP language is - if anything - more difficult than those for dot-matrix printers and quite beyond the comprehension of most users. What Paul was looking for was a ready-made driver that some other reader had written. Maybe he still needs it, so please contact me if you have such a thing.

There are a few comments to make on this. Avoid buying a printer before establishing that it will work with the software you use (or vice-versa). When buying a laser printer, prefer one that has HP LaserJet II emulation built-in, because that is the laser standard, in the same way that Epson FX-80 is the dot-matrix standard. In emulation mode, you will lose some of the printer's capabilities, so do not buy a printer on the basis of test printouts in its "native mode", unless you are sure your software has a driver for that mode. Clearly Perfection (and almost all other QL software) will not have a Panasonic KXP-4430 driver. Most QL software will not have an HPLJ II driver either, so you may be reduced to using the printer in Epson FX-80 emulation mode, as a high-class dot-matrix printer.

Even outside the QL world, you cannot get away from the awful business of writing your own printer-drivers, unless you buy only the most-popular models, or accept limited utilisation of your printer's facilities. Here is an example of what you are presented with when investigating how to get an HPLJ II to print the font you want:

"Font Management, Specify Font ID - ESC*c#D 027 042 099 # 068"where "# is the font ID number, 0 to 32767".

That seems to imply more fonts than any HPLJ I've come across has! What does the average person make of that bit of information? It comes from one of three thick guides with my Epson laser printer. Yes, I think I would be able to do something with it, after more than eight years' dealing with such obscure information, but most average users would not have a clue. Fortunately, when using the Epson with the QL, my task is made simple, because text87 Plus4 has both Epson GQ-5000 and HPLJ II drivers available for it.

David Wyatt wrote to inform us that the scanner he had ordered from TK Computerware had arrived. He was displeased that there was no word of apology for the delay, but he says he is delighted with the scanner itself.

Trading standards

We do not get many complaints about suppliers these days, and those we get are usually settled without too much difficulty. For readers unfortunate enough to have unsettled grievances, the advice is always to contact the Trading Standards Office in the area where the supplier is based; when legal action looks to be desirable, claims should be made through the Small Claims Court at the County Court, again in the locality where the supplier is based. You can obtain information about both through your own local Trading Standards Office and Courts.

INFORMATION
Components:
Greenweld Electronics
27 Park Road
Southampton SO1 3TB
Tel 0703 236363
Fax 0703 236307

Mr M Walkerden
Trading Standards Office
11 Church Street
Folkestone
Kent CT20 1TN
Tel 0303 850294
Fax 0303 221508

Fleet Tactical Command II for the QL £39.93, for the PC £49.95, plus post and packing:

Di-Ren
59 William Street
Walsall
West Midlands WS4 2AX.
Tel 0922 33580

Vector-drawing program:
Dilwyn Jones Computing
41 Bro Emrys
Tal-y-Bont
Bangor
Gwynedd LL57 3YT
Tel 0248 354023

and

Software87
33 Savernake Road
London NW3 2JU.

Trouble Shooter

WIRING OF CONNECTOR PINS AT ENDS OF LINKING CABLE FOR FLEET TACTICAL COMMAND, QL SER1 TO PC COM1 (MALE CONNECTOR AT THE QL END, FEMALE AT THE PC END)

SIGNAL:	QL CONNECTIONS:		PC CONNECTIONS:	
	6-PIN PCC	9-PIN D	9-PIN D	25-PIN D
GROUND	1	1	5	7
TXD	2	2	3	2
RXD	3	3	2	3
DTR	4	4	7	4
CTS	5	5	8	5

TF Services

MINERVA
The ULTIMATE operating system upgrade
MKII MINERVA now with battery for 256 bytes ram, CRASHPROOF clock & interface (I²C) for robots etc. Can autoboot from battery ram.
Other features (common to MK I/MK II):
Autoboot on reset/ Multiple Basic/ Faster scheduler-graphics(10% of Lightning)-string handling etc/ WHEN ERROR/ Second screen/TRACE/ Foreign kybd drivers/ 'Warm' Fast reset etc
MK I to MK II UPGRADES: up to 1.79 - £5. Rest FREE (send rom (PROTECT_LEGS!) and disk or 2 mdvs)
Original MKI..... £40 MKII.....£65
ALL VERSIONS FULLY GOLD CARD COMPATIBLE

HERMES
A replacement QL co-processor (IC 8049 IPC)
Do you get keyboard bounce?
Do you find fast serial input unreliable?
Do you want to run an HST (4400 bps modem?
If you can say YES to any one of these, then this is a product you have been waiting for
- Gives up to 19200 bps RELIABLE serial input
- Independent baud rates
- Stops keyboard 'bounce' (unwanted repeat chrs)
- Improves 'fuzzy' and 'random' sound
- Provides extra input and output lines
- Key click
- Reset/INT7 invoked more safely

FITTING is a simple job. Simply remove the QL top (8 screws) and replace the chip marked '8049' next to microdrive 1.
PRICE (including manual & software) £25

BULLETIN BOARD
BEES - The first UK QL scrolling bulletin board. Now even longer with international ECH(mail und 10mb of files (including the WHOLE King James' bible!)
071-706 2379 V21/23/22/22bis 24hrs

QL SPARES
Faulty QL boards (excl plug in ICs)..... £9
Keyboard membrane... £8 Circuit diags... £2
68008 cpu..... £8 1377 PAL..... £3
JM rom set..... £10 Power supply... £12
8302 ULA..... £10 8301 ULA..... £10
8049..... £8 MDV ULA £12
Other components (sockets etc)... plse PHONE

QualSoft
Viewdata/VT52/VT100 QL TERMINAL EMULATOR
Multitasking for electronic mail, PRESTEL etc, CET download. Compatible with Tandata Qconnect QL modem/modaptor. Use Qconnect RS232C to drive unbuffered modems. Comprehensive A5 manual.
Software (3.5" or mdv) & manual £30
FILE TRANSFER - to transfer text AND programs error free (XMODEM) between computers
Available for IBM compats, ATARI ST & Sinclair QL. Connects to Psion range with RS232C comms link. Software £7.50 per m/c. Cable £10 (name 2 m/cs)

QL REPAIRS
QLs tested with Thorn EMI test rig and ROM software
(MDV hardware extra if required)
£27 - 6 month guarantee

All prices include VAT and postage in the UK
MAIL ORDER ONLY
VISA
12 Bouverie Place, London W2 1RB (tel: 071-724 9853)
SCROLLING BBS FOR ORDERING/QL HELP and FAX - 071-706 2379

Telephone: 0753-888866
Fax 0753-887149
(EEC) W.N. Richardson & Co.
18-21, Misbourne House, Chiltern Hill, Chalfont St.Peter, SL9 8UE.

SINCLAIR QL PRICE LIST MARCH `93
All Prices Include 17.5% VAT

QL Computers
NEW MONITOR £50 CHEAPER
JS COMPLETE QL Computer, PSU, T.V. Lead, PSION V2.35 Software (Word Processor, Spreadsheet, Database & Business Graphics). Handbook for QL, Programs', & Superbasic. Free one years membership to QUANTA, the independant QL User Group with over 400 free library programs, Newsletters & HELP. 6 MONTHS WARRANTY. JS £120 WITH JM ROM £100.00
PART EXCHANGE £30 OFF ABOVE. Send in old QL (Unit only, any condition) - JS ROM £90 JM £ 70.00
BACKUP QLS. QL & PSU only. JS £80 JM£65 (Part exchange allowance £15 if required)

Accessories
* NOTE: EXTERNAL (SER2) 3 BUTTON MOUSE AND SOFTWARE WITH EXTRA FUNCTIONS. NOW HERMES COMPATIBLE *
PC KEYBOARD INTERFACE, INTERNAL FITTING, FOR FITTING 102 KEY KEYBOARD £ 75.00
PC KEYBOARD, UK version, 102 key (AT) £ 30.00
PC KEYBOARD INTERFACE and KEYBOARD (INCLUDES FREE JOYSTICK OR PSU) £ 95.00
CASE and LEAD for EXTERNAL FITTING of Keyboard Interface £ 14.00
JOYSTICK with QL lead (no interface required) £ 10.00
* QL MOUSE, 3 Button, Software controlled, externally mounted, simply fits in SER2 £45 CORDLESS MOUSE £ 55.00

Disk Drives
THE UNIVERSAL DRIVES ARE ALSO SUITABLE FOR ACORN BBC, ATARI ST, AMIGA, SPECTRUMS, IBM, AMSTRAD AND OTHER PC COMPATIBLES
Universal 3.5" 1Mb disk drive, cased with PSU and free QL lead £ 70.00 2 Units for dual drive £120.00
Universal 3.5" 2Mb disk drive, as above but 2Mb capacity £ 90.00 2 Units for dual drive £170.00
3.5" 1Mb Uncased disk drive £ 25.00
3.5" 2Mb Uncased disk drive £ 49.00
Lead for uncased disk drives, or other universal micros £10.00 POWER SUPPLY FOR UNCASED DRIVES £ 6.00
METAL CASES FOR UNCASED DRIVES £ 6.00

Parallel Printers
NEW RANGE
SAMSUNG SP093 80 Col, 300 cps, 50 NLQ, 3K Print Buffer, Centronics, Tractor Feed/Sheet Feed, Paper Park £149.00
OLIVETTI COLOUR PRINTER, 24 pin, 200 cps, 50 NLQ, 40K Buffer, Espon LQ2550/IBM Compatible £259.00
CENTRONICS PARALLEL Printer Interface for above printers £ 27.00
QL LEAD FOR SERIAL PRINTERS £10 CANNON BJ10 EX £225

Monitors
PHILIPS 14" HIGH RES COLOUR EGA .31 DOT PITCH, FULL 85 COL WITH AMBER OR GREEN TEXT FEATURE, IDEAL FOR WORD PROCESSING, RECONDITIONED, 90 DAY WARRANTY. £149
OK FOR PC AND QL SELF SENSING AND MANY EXT/L CONTROLS.
"MONO OPTION" WITH 9" GREEN SCREEN £39

Microdrive Cartridges and Spares
4 New Cartridges in a wallet £ 10.00
Plastic Storage filing box including 20 new cartridges £ 15.00
QL Psion Software V2.35 Includes Quill, Abacus, Archive, and Easel IN WALLET £ 45.00
QL Psion Software Separate programs £ 18.00
8 prog carts for reformating in 2 wallets £ 10.00
QL power Supply Unit £ 10.00 Membrane (and instructions) £ 9.00
TV or Network leads £ 3.00 QL Top & Bottom Case £ 5.00
ICs ZX 8301 £ 6.00 ZX 8302 £ 3.00 8049 (IPC) £ 3.00 MC 1377 £ 3.00
MC 68008 £ 12.00
QL SERVICE MANUALS & CIRCUIT £25.00
S.A.E. FOR FURTHER DETAILS

VISA
Payment terms: CWO, Access, VISA et cetera Cheques - allow 10 days
Delivery: Carriage - £9 Postage - £5
Product subject to availability, E&OE.
TEL: 0753 888866 FAX: 0753 887149

SOFTWARE87 MOVES INTO DTP

Software 87 have launched a modular desktop publishing package which allows their popular *plus4* wordprocessor to be integrated into a full text, graphics and page-layout combination for the QL. The package is recommended for Gold Card QLs or Atari/QL emulators, but will also run on QLs extended with a Miracle/Qubbesoft Trump Card.

The main partner to plus4 in the new system is *Linedesign*, developed by the long-standing Belgian software company Progs. Linedesign allows the creation of pages consisting of scaleable illustrations and text. Once drawn, items can be moved, enlarged or reduced, expanded, squashed, rotated or transforemed without losing detail or affecting other objects on the page. Linedesign can treat text in the same way, combining any of 57 scalable fonts supplied as part of the program.

Linedesign is also notable for comprising no fewer than 11 disks, including the program, disks of scalable fonts, and a large collection of ready-made drawings for use. By itself, Linedesign costs £99 from Software 87.

The link between Linedesign and plus4 is *publisher's pack*, a utility which allows the user to combine the editing and printing capabilities of plus4 with the graphic and font handling of Linedesign. The program comes with a step-by-step instruction manual, several sample documents, and ready-made page-design items such as shadowed boxes. Publishers pack by itself costs £39.

The whole package - the latest version of plus4, Linedesign and publisher's pack - under the name plus4 publisher costs £189. Linedesign and publisher's pack together as an upgrade to existing plus4s costs £119, and the current version of plus4 by itself costs £79.

Software 87 is also carrying a range of the most popular German QL programs, including the latest versions of *QD*, *QSpread* and *Datadesign*. They have details and well-laid-out leaflets on most of their current list - for more information send an SAE to **Software87, 33 Savernake Road, London NW3 2JU.**

plus4 publisher

QL Desktop Publishing and Wordprocessing at its Best

NEWS!

Software87 are pleased to announce an important development on the QL software scene.

Thanks to **plus4**, QL users have been able to produce high quality text output matching that of expensive PC word processors. But, until now, no QL program has been able to produce drawings and large text of similar quality (up to 360 x 360 dots per inch).

The new page design program **LINEdesign** allows the creation of pages consisting of scalable drawings and text. Once you have drawn an object, you can move it around the page, enlarge or reduce it, expand or squash it, rotate or transform it without losing any detail and without affecting other objects on the page.

In addition, **LINEdesign** can do the same with text, using any of the 57 scalable founts supplied with the program. Each fount can be printed in any size from small print to huge headline without loss of quality. **LINEdesign** is supplied on eleven disks which contain the program, the founts and a large number of ready-made drawings (clip-art) for inclusion in your publications.

plus4 and LINEdesign work together

text from **plus4** can be combined with **LINEdesign** output in two different ways. Using one method, text is produced and printed with **plus4**. The page layout is then sent to **LINEdesign** for positioning the graphics and headlines which are then overprinted on the page from **LINEdesign**. The other method allows text to be formatted in **plus4** using the fount supplied with **LINEdesign**. Formatted text, including different fount selections and justification, is sent to **LINEdesign** and printed together with added graphics.

Publisher's pack allows you to combine the text editing and printing capabilities of **plus4** with the graphic and fount handling power of **LINEdesign**.

A step-by-step guide together with several sample documents and ready-made graphic objects for desktop publishing such as shadowed boxes provide all you need to start preparing your own professional quality publications.

plus4 publisher comprises the **plus4** word processor, **LINEdesign**, and **publisher's pack**. A Gold Card or Atari with QL Emulator is highly recommended but the programs will run on QL's with the Trump Card.

QLScene

SHOWN IN THE USA

International QL Report's Bob Dyl in the USA is arranging a one-day QL show and symposium in Newport, RI, on the American east coast. Several UK suppliers are expecting to attend, including Miracle Systems, TF Services and WN Richardson, who will be demonstrating their latest projects. US users and user groups will be showing their latest software developments. The show will be held on June 5 at the Salvation Army Building, 51 Memorial Blvd., Newport RI 02840, USA from 1pm to 6pm. The cost is US$5 on the door, or US$3 if pre-registered.

For more information, **Bob Dyl at Seacoast Services can be reached on (USA) 401 8493805.**

In PhilTration

QL World would like to add that the program featured in Figure five of The Pointer Environment (page 37, QL World, February 1993) was written by Phil Borman of Quanta. Phil confirms that the program is on the Quanta library disk UTILS_GEN4, and that the disk SPECIALS_0, mentioned in the same article as the source of TK2_EXT and LRESPR, is listed as SP_0 in the current library catalogue.

DIY Toolkit rides again!

DIY Toolkit volumes are available once more on disk and microdrive, thanks to former Quanta editor Dr. Bill Fuggle. Prices have been reduced and A5 laser-printed booklets are now available to complement each volume, along with the code, commentary, tasks, SuperBasic and assembler source on disk or tape.

Bill has taken over distribution of the 23-volume set of Qdos accessories developed by Simon N Goodwin over the last six years. Each volume includes all the programs from several connected issues of the long-running QL World column, plus many extras that have been written since or did not fit in the magazine. Recent updates include HD and ED disk checkers for Gold Card, and extensions for Thor and Minerva.

DIY Toolkit volumes now cost a flat rate of three pounds each, on 3.5 or 5.25 inch Qdos disks, or four pounds per volume on microdrive. The original order-processing charge has been eliminated, so prices start at just three pounds for a single disk volume, delivered anywhere in the world. users ordering two or more volumes receive the printed documentation at no extra charge.

Orders by sterling cheque, bank draft or postal order made payable to **DIY Toolkit, 86 Lordswood Road, Harborne, Birmingham B17 9BY, UK.**

Open Channel

Open Channel is where you have the opportunity to voice your opinions in Sinclair QL World. Whether you want to ask for help wityh a technical problem, provide somebody with an answer, or just sound off about something which bothers you, write to: Open Channel, The Blue Barn, Tew Lane, Wootton, Woodstock, OX7 1HA.

Sideways

In March, R S Matthews was asking about sideways printing from Abacus in particular.

One such program, written more or less explicitly for this purpose, was WriteTurn. This was available from Sector Software, who no longer advertise in QL World. The program uses the graphics facility of 9-pin dot matrix printers, and works on a _lis file dumped from Abacus (or any other Ascii file, presumably). A special sideways character font is used as the block of saved text is read from file in the special order needed to print column by column rather than line by line. The length of paper possible is that for 250 characters, while the height (the narrow dimension of A4) can be set to give from 48 to 192 rows. This latter figure gives a rather squashed text, but it is still readable.

I have used the program a lot, with Abacus, for presenting Family Trees which always work out very long horizontally, and not so very deep. In fact they usually go beyond the 250 print columns (not Abacus columns), which means having two consecutive printouts which can be easily achieved by modifying the WRITETURN boot file, and saving two or more _lis files in ram. I normally use printer rolls, but fan-fold paper should work just as easily to get 5-foot printouts.

For this application, Abacus is used in text mode only. This is a little awkward, since text cannot always be placed exactly where you want it - it does not recognise leading spaces. However, because the character font is accessible, I use a dummy character as a space, and then modify this character in the font to be a blank. You can also redefine some of the other little-used characters to create the graphic bits required to draw boxes, and the linking lines necessary for Family Tree pictures.

Peter Tyler
Ormskirk
Lancs.

We haven't heard from Sector Software for quite a while. Has anyone written, or discovered, any other programs that will do a similar task?

Colour

In the interests of not actively encouraging the wearing of dark glasses in the workplace, I wondered if I should suppress the following information. But I thought, no - don't be a spoilsport. At least John only uses one typeface ...

Influenced by John Parkin's article in *j*2June 1991 about the HP Deskjet 500, I decided to retire my old Brother M-1109. I couldn't believe my luck when I was able to buy the very latest Deskjet 550C, which allows for colour text printing as well as graphics, for only £405 + VAT.

Was there a wordprocessor capable of driving colour within a document? Freddy of Digital Precision told me that the 8 strips in Perfection Special Edition would do the trick. Having not received similar assurances about text87, I forked out £100 for DP's processor.

When the new printer arrived I had visions of sending out my marketing materials the next day in glorious technicolour (and you a computer expert too!). Wrong! The printer, I expect, would have worked soon enough if I'd plugged it into a PC loaded with Microsoft Windows (That depends what you mean by 'soon enough'. We've just discovered a well-known Windows program that can't find its own printer-driver.) The users' manual was not very user-friendly. However, after much headscratching and the aid of a technical manual sent from HP in Holland, eight weeks later I have finally done it! (It's true. This last sentence is printed in tasteful green, red, blue, yellow and magenta.)

The printer driver which accompanies this letter might save readers seven weeks, six days and twenty-three hours.

Note: the symbol '&' has to be entered using the decimal equivalent '&38'. I ran out of strips, and the code for colour cyan is <ESC>*V2S. Can't have everything! The Configurator in Perfection SE can be used to reproduce the colours on the screen. Translate 1 gives me the £ symbol using the PC8 character set. Translate 2 gives me normal-size CGTimes typeface, which I prefer. As it is proportional, I need to set my margins at 15 on the left and 115 on the right for normal A4 paper. This produces a pleasing ragged right. Translate 3 switches on the four-plane CYMK palette for colour printing. Translate 4 switches on the single-plane palette for black-only printing. Translate 5 switches on the colour black. Translate 6 switches on smaller CG Times typeface. So, if I wish to print a colour document in normal-size CG Times I put the translate 4 and 2 symbols in the top left corner of the screen. Ctrl-Shift-K will print the symbol 1/2_, and Ctrl-Shift-L [we think this is an L, but it could be a figure 1. John's typeface is genuinely indistinguishable] prints the symbol 1/4 _.

I have only two problems to overcome: currently I have to switch off the colour at the

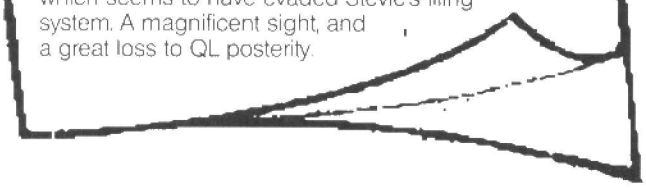
Underline On: <ESC>&d1D
Underline Off: <ESC>&d@
Bold On: <ESC>(s3B
Bold Off: <ESC>(s0B
Superscript On: <ESC>(s6v+1U
Subscript On: <ESC>(s6v-1U
Italic On: <ESC>(s1S
Superscript Off: <ESC>(3@<ESC>(s+0U
Subscript Off: <ESC>(3@<ESC>(s+0U
Italic Off: <ESC>(s0S
End of Line: <CR>
Initialisation
End of Page: <ESC>&10H
Strip 1 (Highlight Off): <ESC>(s1p12v4101T
Strip 2 (Highlight On): <ESC>(s1p14v4101T
Strip 3 (Colour red): <ESC>*v12S
Strip 4 (Colour green): <ESC>*v10S
Strip 4 (Colour blue): <ESC>*v6S
Strip 5 (Colour yellow): <ESC>*v8S
Strip 7 (Colour magenta): <ESC>*v4S
Strip 8 (Colour off): <ESC>*v1S
Translate 1: £
Translate 2: ^<ESC>(s1p12v4101T(Keys SHIFT 6)
Translate 3: "<ESC>*r-4U<ESC>*v12S (Keys SHIFT £)
Translate 4: Ø<ESC>*r1U(Keys CONTROL SHIFT R)
Translate 5: µ<ESC>*v1S(Keys CONTROL SHIFT P)
Translate 6: ¤<ESC>(s1p10v410T (Keys CONTROL SHIFT W)

Editor's notebook

Anniversary! It's exactly one year since QL World fled the dust and noise of London Town and came to join Arcwind Publications in the green fields of Oxfordshire. The Oilseed flowers are blooming again. It's exactly one year and six months since Robert Maxwell fell overboard and we thought (not for the first time): "Now what?" We're still here.

As I pointed out to a fine mathematical mind recently, "Volume II: Issue 4" means, "Our second year, fourth month", and reflects our general embarrassment at the lateness of recent issues. I may try to produce two issues close together at some stage if the production team can stand it, but I can't work our authors any faster. Talking of that, what's happened to Mike? He was going to send me his copy last week. I know he's a busy man, but ...

The mysterious reference to Bill Newell doing his bit for PR in last month's Eindhoven report will become clear if we can find the missing photo, which seems to have evaded Stevie's filing system. A magnificent sight, and a great loss to QL posterity.



beginning of each line of black text to prevent unwanted magenta appearing. Also I have not yet found an easy way of producing proportional-spaced text in columns.

Now wouldn't it be great if someone devised a printer-driver for Easel so that I can print coloured bar and pie charts? I can dream, can't I?

John Gooding
John Gooding and Associates
Marlow-on-Thames

Decoded

Having read *Beginners Machine Code* in the March issue, may I point out a few small errors in the text that could be a source of problems for the uninitiated.

There are only 8 data registers (D0 to D7) and only 8 address registers (A0 to A7) not ten as stated.

The DBRA or DBF instruction does not stop when the appropriate data register reaches zero, it actually stops when the data register reaches -1. For this reason, the actual value stored in the data register must be 1 less than the number of cycles around the loop required.

In the article, listing one shows the value 8000 (hex) being loaded into D0 in order to display a striped pattern on the screen. The following loop will actually overwrite the first byte of the system variables following the end of the screen data. The correct value should have been 7FFF (hex).

At various places in the text, references to the loop counter reaching zero should be replaced by -1. I think that Alan does know about the DBRA instruction stopping at -1. otherwise why would he store zero in register D0 at the end of listing one? If the loop counter stops at zero, then in the example, D0 would have zero in it.

I hope this clears up any problems beginners may be having. I know when I first started assemly language programming many years ago I always got this wrong and expected loops to stop when the counter reached zero, due to my background in Z80 assembler, where the loops actually do terminate at zero.

Norman Dunbar
Bucksburn
Aberdeen

Alan Bridewell replies: Oops. This is what happens when you deal with similar instructions on different processors, like the Z80 and the 68000, and forget to check the programmer's reference manual because you think you know what you're doing. If the mistake was likely to cause a crash, I would very likely have spotted it. As it happens, the listing works as printed, so whatever is being overwritten in the system variables isn't very significant in this case. I'm afraid the same mistake has occurred in part 3, which has already gone into the printing process, but once again it should have no effect of operation in this listing.

As to the other point, I blame my wordprocessor. It simply won't learn that when you count from 0 to 7, you need 8 fingers, not 10. My humble apologies on our behalf.

End of the line

This is not an April fool. Since last summer I have had access to a 386DX PC with *Timeworks* 3.00 (Windows) at a local printing resource group. I produce our church magazine there. When I transfer my documents, typed in on *Turboquill+*, tidied with *TexTidy* and transferred by DiscOver, I always lost paragraphing in Timeworks. For MS-Works I lose line settings.

Imagine my joy at Simon Goodwin's listing on page 36 of the March *QL World*. Imagine my frustration at having sat up till midnight typing it in and trying to get it to work, only to find that it did not work! When I ran it it came up with the following: At line 250 end of file. Line 250 read INPUT #3,a$. I rechecked my listing and it was right. What gremlins are these?

Hold everything! I've discovered the problem. It lies with the end-of-line codes set in TexTidy. If they are set to other than LF (is none, CR/LF, or CR) then Simon's program crashes. Set TexTidy to LF as your end-of-line code when making the .txt file and all is well.

James McGreehin
Alva

End-of-line codes have proved to be absolutely the biggest bugbear on converting programs. This is off James's point, but one hint for people sending Ascii-converted files. The worst problems arise because the destination program invariably can't tell (what were once) the wordwrap returns from the hard carriage returns. The solution is usually simple: ALWAYS leave an empty line between paragraphs - this immediately provides a consistent double-return code to distinguish the paragraph boundaries. Avoid para indents - they don't provide a readable code, and can actually confuse the destination program.

There should be an end-of-sub code on your address label, but if in doubt ring Pauline and ask her.

Error trapping

I have been a QL dabbler since 1988, 'still white behind the knees'. I am more than pleased to have a magazine other than Quanta publishing bits and pieces about 'our' micro which are of interest to everybody. I write programs mainly for my own and my family's interest. As you have asked for suggestions for articles, I would like to ask for an article or short series on the subject of writing SuperBasic modules on error trapping of input to be included in interactive programs. As an example, inputting a value between -5 and 5. This is straightforward with a numeric entry such as IF n <-5 OR n>5 , but what if 'k' is input in error? I do have disks and a Gold Card, so I have the use of Toolkit2. I am sure others would find this useful as well. Keep up the good work.

R D Hardie
Great Yarmouth
Norfolk

Simon??

Rocky mods

I have owned a QL for eight years now, and have almost every issue of QL User and QL World. Recently, I was given a second QL. It is an issue D07, and despite carrying out the Sinclair modifications to it, it will

not work with a different rom set. It currently has 2 x 16K eprom in IC33 and a 16K eprom in IC34. There is also a flying lead from the 'piggybacked' eprom to the circuit board alongside the 8302. There is also a socket containing IC17. Having tried my D13 JM roms, a 64K MG eprom and my 1.64 Minerva eprom, with and without IC17 in place, without any success, I am writing to you to ask if any readers have any other ideas or modifications to be carried out? The mods carried out so far are:

R104 (82 ohms) in TR1 collector circuit
R92 (220 ohms) changed to 390 ohm
R105/R106 (1 kohm) across C19 and D17
D22/D23 (1N4148) in series with R100/R101
R102/R103 (33 kohms) between IC23 pins 21, 19 and -12V.

Then the firmware can be upgraded to build standard D11. I known that the Sinclair manual has some inaccuracies, and I can only assume that there is something wrong with the mods.

Also, my master and backup copies of ICE have been corrupted. If anyone can let me have a copy I would be grateful. And, lastly, if there is anyone in the Lincoln area who would like to correspond, please write to me via QL World. Thank you.

Ian Turner
Waddington
Lincoln

As I read this, the names 'Tony Firshman' (TF Services) and 'Dennis Briggs' (Adman Services) kept repeating themselves in my mind. And, of course, there are plenty of other knights of the soldering iron out there who have done these mods. Also, anyone who wants a pen-QL, don't pass up this offer. Too many QL users still operate in virtual isolation. Make a contact.

List puzzle

I obtain ribbons for my Seikosha printer from Imac Ltd., Stuart Road, Manor Park, Runcorn, Cheshire WA7 1TH. Tel. 0928 579000. Imac UK also produce a very good catalogue of office and computer supplies, with a good delivery service. There is a small order surcharge for less than a pack of six ribbons.

May I say how pleased I am that you are including Simon Goodwin's SuperBasic programs in QL World. I am sure they will be of interest to many readers and my own South West Quanta User Group.

There is something which has puzzled me for quite a while, which has not been explained to me by my associates or by a question to Alan Bridewell at the Bristol Workshop in Clevedon. It is regarding listings of the type of Listing one on page 36 of the October 1992 QL World.

I have been a subscriber to QL World since 1986, and have enjoyed the many items of interest including successfully typing in the SuperBasic programs in the magazine. Who knows, I may become a program writer yet, and send in a program myself!

Roy H Stephens
Wellington
Somerset

When I first asked Alan to do a Beginners' Machine Code series, he said it was difficult to explain! The listing you mention is one of Rich Mellor's, so I'll give him a crack at it. Watch this space.

Endless loop

L Atkins *(Open Channel February* 1993) seems to have had problems with my FONT routine - I do not know how the double $$ marks managed to get into the listing. they were not there before! If he needs any more help with that (or machine code) he can contact me.

I can supply the latest version of *Qram* to Laurence Carpenter if he sends me (via *QL World)* his original mdv, a backup mdv, an SAE and £2.

To J R Daniels (March 1993): lines 450 and 460 are rather unconventional programming style, but simply loop onto themselves, reducing (or increasing) y by two pi, until y<two pi and y> respectively. They could be replaced by the (slightly slower):

```
450 REPeat cut_y:IF y<=two pi:EXIT cut_y:ELSE y=y-two pi
460 REPeat inc_y:IF y>=0:EXIT inc_y:ELSE y=y+two pi
```

Rich Mellor
Walsall

Abacus bug

Regarding Peter Jones' letter regarding the oddity with Abacus where the displayed spare memory is immediately followed by the cell contents, this is a bug in Abacus which manifests when the spare memory exceeds 999K. The only fix is to reduce the memory available to Abacus.

The memory available to *Abacus* can be reduced by using Simon Goodwin's *Taskforce* program listed in a previous *QL World,* or simply by taking the memory for something else prior to executing Abacus, for example by typing:

```
RES_SIZE 1216
```

which will reset the QL to 1216K (note the number of K must be a multiple of 64). After the reset run the Abacus boot program. Alternatively, add the following line to the start of the Abacus boot program:

```
FORMAT "RAM1_1250"
```

which will create a useful ramdisk with the memory not used by Abacus.

Stuart Honeyball
Miracle Systems
York

Networking

I really appreciate the *DIY Toolkit* articles by Simon Goodwin and his predecessors. I use them to enlarge my knowledge of how computers work. Although I am far from being an expert, I am fascinated by the methods demonstrated in Simon's articles.

The setup that I use is based on two networking QLs. On one (QL1) I run the machine code of the DIY SuperBasic extension that I am studying. Its progress is traced by using Qjump's QL-Monitor. Meanwhile, on the other (QL2) I have Taskmaster controlling the multitasking of IDIS Disassembler, GST QL-Assembler, some small utility programs and several copies of *Editor Special Edition.* Into one of the latter, I load the assembly code for the extension, and then add copious comments to it as I follow the machine code on QL1.

One of the advantages of using the network and the Monitor program is that I can transfer the contents of the registers and memory from QL1 into the assembly code text in QL2, where it can be converted into comments that help me to understand what is going on.

Often when I am 'messing about' with the machine code, I cause QL1 to hang up. Resetting QL1 has no effect on QL2. This is another advantage of networking.

At present I am trying to follow Simon's FlexyNet code. This is not easy because it disables the interrupts. This stops Monitor from being able to trace. Thus, I have to advance the program counter over the disabling step. Perhaps FlexyNet will still work if I slow it down enough (probably not ...)

The more I study Qdos, even though it is beginning to creak with age a little, the more I suspect that we in this country may have once more missed an opportunity to develop a brilliant idea. Where would personal computing be now if Qdos had had the investments that MS-DOS has had? Possibly many completely memory-based fast packages, each using common multitasking utility facilities. Perhaps no need for hard disks.

Still, that is just dreaming. In the meantime, I hope that Simon will continue supplying the material which will give me hours of future entertainment.

David Williams
Withersea
North Humberside

Open Channel

QLScene

Merz Goes Game

Jochen Merz has released two new games for the QL. The word may have gone round that the QL is no game-player's machine, but "serious" games still have a healthy following. *The Oracle* is billed as a "game for QDOS and the Pointer Environment", a "beautiful and ancient game our of the depths of history ..." You must boot up with a clean machine, repeat the filename twice, and mediate carefully over each move. The first correct solution to reach Jochen before the end of December 1993 wins a prize.

The game is played by placing coloured tiles on a 96-square board, creating as many matches as possible. The centre of the board is called the Within, and the outer edge is called the Beyond. The Oracle speaks and gives you hints whenever you complete a high-scoring match.

People who have played an easier version of this game say that the secret is to set up as many opportunities for a match as possible, and also that it is addictive. Perhaps Jochen has discovered the ancient secret of game-playing on the QL.

The other game, *MineField,* though less lofty in inspiration, also uses the Pointer Environment (which is included on the disk), and Toolkit 2 (which isn't). The program supports the higher resolutions available to the QL Atari emulator. The game consists of uncovering all the squares in a field without being eliminated by a 'mine'. The instructions contain a (very) short intorduction to the Pointer Environment.

The Oracle is Jochen's own work, and he is the distributor for MineField. We have no price details yet, but watch this space. More information from **Jochen Merz Software, Im Stillen Winkel 12, DW-4100 Dulsberg 11, Germany. Tel. (Germany) 0203 501274.**

Now check it again

After much discussion and several telephone calls between the editor, Simon Goodwin and office manager Fran Maxwell to establish the correct version, we printed last month some missing characters from line 850, listing two of March's DIY Toolkit.

Unfortunately, we printed the incorrect version. The missing characters should be "8018", not "8118".

Commenting shortly after publication, Mr Goodwin said first "eek!" and then "whah!". The Editor would only add "aargh!".

GIVE US A LIGHT

NESQLUG are still alive and well and have been asking editors of other QL fanzines if they might be suffering from editorial burnout (apparently a subject that had been raised within their own walls a bit back). The replies seem to add up to a resounding No!

They were kind enough not to ask QL World, but this editor can report: the only editorial burnout around here is when two (or more) QL luminaries are arguing over (or even at) her head, and then it usually takes the form of bursting into flames.

Meanwhile, in the latest NESQLUG Newsletter, Hugh H Howie is doing the fiery bit.

Keep burning, NESQLUG. Other wandering sojurners seeking guidance from the Abyss of Confusion can reach them at **PO Box 8763, Boston, MA 02114, USA.**

Mouse Pointer for Fido

Erik Slagter of Qdos Software in The Netherlands is offering Qpoint, a mouse-driven Qbox Fidonet mail reader for Qdos, SMSQ and Minerva users. Fidonet use is a specialised area, but gives access to other users all over the world. Erik's best advice for new or perplexed users is to find a current user in your local group or area and ask for their help. But there is some information in the leaflet and order form which he has printed.

The program for current Qbox users costs 40 Dutch guilders or equivalent, plus carriage. The leaflet lists special features and methods of ordering/payment. Get the leaflet by contacting Erik Slagter at **Beljerland 15, 2716 CM Zoetermeer, The Netherlands. Tel. 010 31 79 213659. Fidonet: Erik Slagter 2:281/503.2 or Internet: v882022@si.hs.nl.**

Belgium by fax

PROGS of Belgiun now have a fax number: both voice and fax operate on **+32 16 488952.**

Recent upgrades include DataDesign to 3.04, with new commands in Merge file, Stuff record, View file, and Replace, and some bug fixes; LineDeisgn to 10.3, with bug fixes and higher speed. Current users who have been in touch have recieved free upgrades; other users should contact Progs at **Haachtstraat 92, 3020 Veltem, Belgium.**

QXL GOES PUBLIC

Miracle Systems' long-awaited QXL card - an internal card for MS-DOS, PC-compatible computers which effectively installs a high-specification QL inside PC hardware - is due on the market this month. As we write, Miracle are only awaiting the final touches to the first release software, an entirely rewritten version of QDOS, now known as SMSQ.

Stuart Honeyball of Miracle told *QL World* that the software, being written by QL guru Tony Tebby, is now very close indeed to completion, and that they expect to be shipping by the time the May issue of QL World comes out (currently late May or early June). The QXL will be issued in four memory sizes, giving users the maximum choice and flexibility. These will be 1 megabyte at around £295, 2MB at around £325, 5MB at around £410 and 8MB at around £495.

Miracle already have a good number of orders for the QXL (they won't say how many!) and, Stuart tells us, orders their components stock many months ago, so that there shouldn't be any order backlog. Miracle will also be issuing software upgrades free of charge from the start, so that early customers will not be at a disadvantage.

Stuart thinks that many of their current orders come from people who have discussed the system with him at the computer shows where Miracle make regular appearances. The obvious attraction at the moment is to current PC owners, but he points out that the card could be a route to moving QDOS software into workplaces that it doesn't currently reach, as it can be fitted in virtually any standard AT or XT PC clone. There are three main attractions: getting two computers into one; from the PC's point of view, getting access to a new suite of programs under a new operating system; and it is also a speed upgrade. Once the software is developed to handle SVGA, it'll be a graphics upgrade as well.

The release version of the software, known as SMSQ, is a completely rewirtten operating system with multitasking Basic, developed from Tony Tebby's QDOS rewrite SMS2, and tailored for QL compatibiity. The functionality, says Stuart, will be the equivalent of a QL fitted with a Gold Card, but faster.

Jumping the gun a bit further, what will be happening now on Miracle's other projects? Miracle have finished their work on the QXL hardware, and are now looking to complete the much-desired Graphics Card in the next few months. "The chip has been designed, and now the final 10% of the work has to be done," says Stuart. When the Graphics Card is under way, Miracle will turn again to the SCSI disk interface which they have long been planning.

One of Miracle's projects this year will be to visit the one-day American QL Symposium in Newport, RI, USA on June 5th, along with a number of other European developers and US suppliers. Miracle will be demonstrating the QXL card, as they did in Eindhoven in February.

Miracle will also be going to Ireland for a show in Newtown Abbey (5 miles north of Belfast) from 1am on 26th June. For more details of that event, contact Keith Johnston, 21 Orangefield Road, Belfast BT5 6DB. They also intend to visit, and to set up with Tony Firshman's help, more user-group meetings, instead of fairs. Says Stuart: "We'll consider going anywhere in the world."

For more details of the QXL card, see Miracle's advertising in QL World. Miracle are loaning QL World a unit for photography - another tribute to the company's increasingly streamlined production capabilities.

ALL FORMATS DIARY

Coming dates for the All Formats Computer Fair are: May 22 London Sandown Park, Esher, Surrey May 29 Leicester, De Montfort Hall, Granville Road 30 May West Midlands National Motorcycle Museum, J6 M42 19 June London Novotel, Hammersmith 20 June Brunel Centre, Templemeads Station, Bristol, 26 June North West Haydock Park Racecourse J23 M6 27 June Midlands National Motorcycle Museum, J6 M42, 25 July ditto, 11 September ditto. Either they haven't filled in their schedule yet, or they're off on their summer holidays. There's a full September schedule.

Check with suppliers whether they will be at a particular Fair. If you have far to travel phone All Formats 0608 663820 to check arrangements haven't changed. Many QL suppliers only attend the Glasgow and London fairs. In London the Hammersmith venue is favoured.

Day tickets are £4; attendees can get up to 50 £1-off vouchers by sending an SAE to the organisers at: **Maple Leaf, Stretton-on-Fosse, Moreton-In March, Gloucestershire GL56 9QX.** (Only one voucher per ticket.) Photocopies of these vouchers are also accepted. Admission to the Fairs is a flat £2 between 2pm and 4pm (£1-off vouchers do not apply at these times).

What next? A Gold Card?

QL Scene

An Irish postie has taken up his QL again after years away, all because of QL World's see-through postal packets.

Royal Mail employee James Caulfield put his QL in the cupboard when he decided to try a PC several years ago. But the wonder-machine clearly didn't drive the QL completely from his life, for in February this year the words "Dilwyn Jones Computing - QL Software", on the back of a QL World speeding to its subscriber, caught his eye. Clearly delighted by Dilwyn's list, he wrote to DJC expressing his desire to resurrect his QL. He also asked for a catalogue, and a contact address for QL World.

We don't know if Mr. Caulfield's QL is being cared for by a dedicated QL leprachaun, but we notice that he lives in Sinclair Dell!

Dilwyn Jones prints to the screen and gives some sample programs to type in.

VERY BASIC SUPERBASIC Part 2

In the first instalment we looked at how to type in a SuperBasic program and explained some common jargon. In this second part I'll look at printing commands, give some short example programs for you to type in, and some guidelines on how to debug them if they don't perform as they were meant to!

PRINT is the instruction which tells the computer to write something on the screen, to a printer, or even to a file on a disk or microdrive cartridge. Using the PRINT command to write to a printer or to a file on a disk or cartridge is more complex and will be covered later.

Fussy logic

As we discovered in part 1, the PRINT command is a little fussy about whether we print numbers or text, for example. We have to use numbers, not words, to print numbers. The reason for this is that the QL can use "words" as variable names. So when we tell the QL to PRINT 7, it's quite obvious what we mean. But if we told it to

PRINT dozen

all it does is to print up an asterisk on the screen to tell you it doesn't understand "dozen". Why? Because it needs to be told what the word "dozen" means. You will remember using statements like LET x = 6 from your mathematics lessons - the QL can do algebra too. We use the LET statement on the QL to tell it that the word (or "variable name") following is to *have the value* shown. Enter this command (there must be a space after the word LET)

LET dozen = 12

The variable "dozen" has been given a value of 12, so if we enter the PRINT dozen command again, it now knows what we mean and prints up 12 on the screen. We can change the value of "dozen" in the same way:

LET dozen = 13

If we now print its value again, it will write 13 up on the screen. Now we see why it's called a "variable" - its value can be varied. The computer does not make the same assumptions that we make!

This is not very useful. Let's look at how to change its position on the screen - you might like to type this in as a program rather than the immediate commands I list here, so type the program in figure one, run it and then try to make changes to become familiar with what can be done. It introduces another new command, AT, which positions printing anywhere in one of the "windows" on a screen. When you switch on a QL, and press F1 or F2 to start up, the screen is divided into three 'boxes' (known as 'windows').

Figure one

```
100 CLS
110 LET dozen = 12
120 AT 5,0
130 PRINT dozen
```

Normally, printing commands send their outputs to the red window, "window #1" (the # symbol is called Hash, and here stands for 'number'). The black box at the bottom, where anything you type normally appears, is called "window #0" and the other window (which may be white or blue depending on whether you are using a monitor or television display) usually shows the program you have typed in.

AT command

The AT command specifies where the next printed item will appear in the window to which you refer. The example in figure one (AT 5,0) means that printing will occur 5 lines down, 0 characters across (that is, at the left margin). The values used start from 0 for the top left corner for both distance across and down.

If we want the printing to occur in one of the other windows, you must specify this by using a #number parameter, which is not as difficult as it sounds! Basically, this involves adding a #0, #1, or #2 to some of the commands. Figure two shows how the program looks - you need to add #0, #1, or #2 followed by a comma or semicolon. The significance of punctuation in print commands

LISTING 1

```
100 REMark simple price calculator
110 CLS
120 INPUT"Enter price per item ';
    unit_price
130 INPUT"Enter number of items ';items
140 REMark the \ symbol prints a blank
    line
150 PRINT\'Total price is ';items *
    unit_price
```

LISTING 2

```
100 REM setting the QL clock and showing
    the time
110 CLS
120 INPUT"Enter year (e.g. 1993):';yr
130 INPUT"Enter month (e.g. 4):';mth
140 INPUT"Enter day (e.g. 15):';dy
150 INPUT"Enter hour (e.g. 14):';hr
160 INPUT"Enter minute (e.g. 30):';mn
170 INPUT"Second (e.g. 59):';sc
180 SDATE yr,mth,dy,hr,mn,sc
190 REMark show the time as a digital
    clock
200 REMark use BREAK (CTRL SPACE) to
    stop this program
210 REPeat show_time
220   AT 7,0 : PRINT DATE$
230 END REPeat show_time
```

LISTING 3

```
100 REMark keyboard familiarisation -
    stop with BREAK
110 LET k$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ
   abcdefghijklmnopqrstuvwxyz1234567890"
120 REPeat typing
130   LET a$ = k$(RND(1 TO 36))
140   CLS : AT 5,5
150   INPUT 'Enter this character:'!
      (a$)!e$
160   IF a$ = e$ THEN
170     PRINT \ TO 5;'Correct!'
180   ELSE
190     PRINT \ TO 5;'Wrong!'
200   END IF
210   PAUSE 100
220 END REPeat typing
```

will be discussed later.

The program in figure two will clear the part of the screen normally used to list the program and print the results of the program there. To get the program back, simply type in the command LIST which will display it in the normal part of the screen. LIST can also have a "channel number" (another name for these #numbers) - for example, LIST #1 sends a copy of the program to the red window on the screen.

Figure two

```
100 CLS #2
110 LET dozen = 12
120 AT #2,5,0
130 PRINT #2,dozen
```

Help!

So something went wrong! Let us try to work out what went wrong, or 'debug' the program. Examples of things going wrong might be:

1. An asterisk is printed instead of the value of "dozen". This means that you have mistyped the word "dozen" in either line 110 or 130. They must be the same. Edit the offending line as described last month with the EDIT command! It is also possible that you may have omitted a space between LET and Dozen. The space is essential, and although it is not essential between AT or PRINT and the #2, it is best included since it makes matters clearer.

2. Printing does not occur in the expected part of the screen, or the wrong part is cleared. You have made a mistake with the channel numbers. Check that they are all the same and edit any errors.

In addition to variables and numbers, the PRINT command can also make use of strings. These contain text and are printed exactly as defined. They can be "string constants" (text in quotes) or "string variables" similar to variables described above. Try these commands:

```
PRINT "Hello"

LET a$ = 'Hello' : PRINT a$

LET b = 'Name'
```

The first two work fine; the third fails and the computer prints up the message 'error in expression'. The problem is that if a variable is to contain text, it must have a name ending with the '$' symbol. So to get the third example to work, we should have written it like this:

```
LET b$ = 'Name'
```

This now works fine (we hope!). Variable names must start with a letter or underline character, and may include numbers as well, but must not start with a number, and or be the same as a keyword built into the QL (which would cause confusion). So these variable names would be fine:

```
eggs    _field    w123
```

But these would not because they don't follow the rules:

```
123w    starts with a number
print   same as the keyword PRINT
$dd     starts with an illegal character
co&w    contains an illegal character
```

Three variables

There are three types of variable, consisting of one string (text, etc.) type and two numeric types. The string type holds text and all sorts of characters and has a name ending with '$', while 'integer' types hold whole numbers with values from about -32000 up to about +32000 and have names ending with the '%' symbol and 'floating point' variables (those with names ending with anything other than $ or %) can hold decimal numbers up to a very high value.

Therefore, these statements are valid:

```
LET a$ = 'Hello'
LET b% = 10
LET c = 5.23
```

But these are not:

```
LET c = 'Hello'
LET b% = 'Goodbye'
```

These can under some circumstances be legal, but not a good idea. The QL can sometimes convert strings to numbers if the meaning is clear:

```
LET d = '1234.56'
LET e = a$
```

The second one may fail if the variable a$ does not contain something which can be turned into a valid number (the term for this process is 'coercion' forcing a string to turn into a number if possible).

After typing all those examples in, you may be annoyed to discover that in most cases LET can be omitted! It is legal to type in a command such as:

```
a$ = 'Hello'
```

The computer will usually realise what you mean and deduce that the word LET is intended even if not actually typed! It is a good idea to use it in the early days, since it is a little clearer if included.

```
INPUT
```

One other useful little command is INPUT. It is a real pain and very limiting to only assign values to variables using LET. It would be much easier and more flexible if the computer would allow us to enter a number each time the program is run. INPUT makes the computer pause and ask us to enter text or numbers. For example, "INPUT a$" would make a cursor flash in the red window, inviting us to enter some text (it knows to expect text or numbers from the name, which in this case ends with $ for text). So we can write a small program to allow the values to be entered and printed based on the above examples (see figure three).

Figure three

```
100 CLS
110 AT 7,3
120 INPUT a$
130 PRINT "The text you entered was ';a$
```

Figure three also introduces us to joining together bits to be printed, by putting semi-colons between items in the "print list". Various punctuation symbols can be used between such items:

```
;  Semi colon - print next item immediately after previous one where possible
,  Space items out in columns or zones on the screen
\  Print next item at start of next line (on the left)
!  If the next item will fit on the same line, print a space between the two items and print the second item on the same line, or if it doesn't fit just move to a new line before printing.
```

TO This little word is normally followed by a number to indicate how far across the screen to move to (hence the name) before printing the following item. It prints enough spaces to reach the column specified. If it is already past the column specified (columns are counted across the screen, rows are calculated downward) it will usually print one space before realising it has already gone past the place specified and print the next item there!

Using INPUT in the way shown in figure three is all well and good, but what is the point of just having a flashing cursor appear on the screen with no hint of what to do? We can either add a PRINT command to display some instructions, or we can use INPUT's other little trick - it can print text like a PRINT command. Figure four shows how to do this.

Note that there must be a semi-colon (or other punctuation) between the variable name a$ and the text (which must be in quotes, although you can use double or single quotes " or ').

Exercise

Modify the program in figure four to enter several items, some text and some numbers. Use different text for each prompt but make sure that the items are entered in a column beneath each other (hint: use a TO statement such as INPUT "text" TO 20;a$). You will need a few variable names of course. Experiment with long PRINT and INPUT statements consisting of several items to enter and print all items on one line.

Figure four

```
100 CLS
110 AT 7,3
120 INPUT "Please enter some text ";a$
130 PRINT "The text you entered was ';a$
```

Multiple statements: all programs so far have simply had one command per line. It is possible to join several together

on one line (though it is not good practice to make long lines like this) by putting a colon between each item. For example:

```
100 CLS #0 : AT #0,2,0 :
PRINT #0,"This is window #0"
```

Listings

Let us now move on to a few simple example program listings. Listing one acts as a simple shop calculator, where you are asked to enter a price for one item, then the number of items sold and displays the total price. Please note: when entering a price, do not enter a currency unit such as $ or £, as it will confuse the program and cause an error. The simplest way of allowing the currency unit to be displayed is to add it to the end of the text after the input statement, something for you to work out how to do! Note the REMark statement at the start of the program - this is simple a comment or remark to help you remember what the program does. Anything after the REMark statement is ignored by the computer, it is just there to remind you. The * symbol in the print statement is the computer's 'multiply' symbol showing you want it to multiply the two numbers in the variables together before printing them.

Listing two shows how to set the QL's built-in clock and show it continuously as a digital clock display on the screen. The SDATE command is used to set the six parts of the time (year, month, day, hour, minute and seconds) and the DATE$ command is used to display the time and date in standard QL format. The REPeat and END REPeat commands are used to make the program execute a part of itself continuously and will be dealt with in a later instalment. Briefly, REPeat carries on doing the statements between REPeat and END REPeat until you press Ctrl-Space to break out of it. The variable name "show_time" is simply a means of identifying the loop, not important here, but vital in large programs where there may be several REPeat statements!

Saving

At this point, we must learn how to save these programs on disk or cartridge for future use, especially the ones which have some degree of usefulness!

We save these programs by deciding on a name for them and using a SAVE command. The rules for a name are the same as those for a variable name (must start with a letter or underline and may include underlines and numbers - underlines are used because it is not practical to use spaces in most cases because the computer would not know if the space was a gap between this command and the next or a part of the name). Names can be used in quote marks or without quote marks and should start with the name of the drive on which they are to be saved (mdv1_ or mdv2_ for microdrives, FLP1_ or FLP2_ for floppy disks, for example). It is often a good idea to make the name end with the same three or four characters to allow the files of a particular type to be spotted at a glance when you look at the content of a disk or cartridge with the DIR FLP1_ or DIR MDV1_ commands. I normally save my Basic programs with names ending with '_bas' (which is short for Basic of course). So these might be saved like this:

```
SAVE mdv1_PRICES_bas

or

SAVE 'MDV1_prices_bas'

SAVE mdv1_TIME_bas

or

SAVE 'MDV1_TIME_bas'
```

Loading

To use them again in future, we use the LOAD command, followed by a filename, eg LOAD mdv1_PRICES_bas. If you put the filename in quote marks, you can use characters which cannot normally be used, though this is not good practice!

Listing three contains a simple typing tutor which displays a letter or number on the screen and asks you to copy it. Once you have found the key on the keyboard, type it in and press Enter. The computer will tell you if you got it right or wrong. Since it asks for either lower case or upper case letters, you must use Shift when it asks for a capital letter or it will say you got it wrong! This program is obviously of no use if you are already familiar with the keyboard layout, but if you spend a lot of time looking for a character on the keyboard, using this program will teach you quite quickly where everything is through trial and error. Notice how it uses an IF ... ELSE ... THEN construct to make the decision if you are right or wrong by comparing the character displayed with that entered to see if they are equal to each other. The keyword RND is used to select a character from k$ at random. There are 62 characters in k$ (lower case letters plus upper case letters plus numbers) and the RND statement simply picks a character from 1 to 62 in the string.

String slicing

You can pick out a given part of a string by putting the positions in brackets after the string, e.g. LET a$ = b$(3 TO 6). This is called string slicing, another topic to be discussed in a later issue. The PAUSE command used is simply a delay to allow a message to be shown for a given time (units are one-fiftieth of a second, so pause 100 is for 2 seconds). Note the INPUT statement which displays one variable by enclosing it in brackets to stop INPUT asking you to enter it instead of displaying it, and asks you to enter another variable. This is a good example of the power of QL SuperBasic allowing you to do several complex things in one fairly short statement.

Type in, use and modify these programs to get the maximum benefit from them to learn about Basic. Although there are some parts in them which contain elements we have not yet discussed, they should be fairly obvious especially if you study the user guide at the same time - you might find yourself making progress faster than these articles!

QUANTA

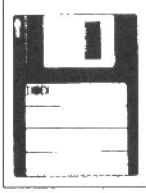
Independent QL/Thor Users Group

Worldwide Membership is by subscription only, and offers the following benefits:
Monthly Newsletter – up to 40 pages
Massive Software Library – mostly FREE!
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software & Hardware problems
Discounts from most major suppliers
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

* Now in our NINTH successful year *

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex. SS7 4JD
Tel (0268) 754 407

Robin Stevenson takes the Common or Garden Average and puts it to work.

Archive Answers

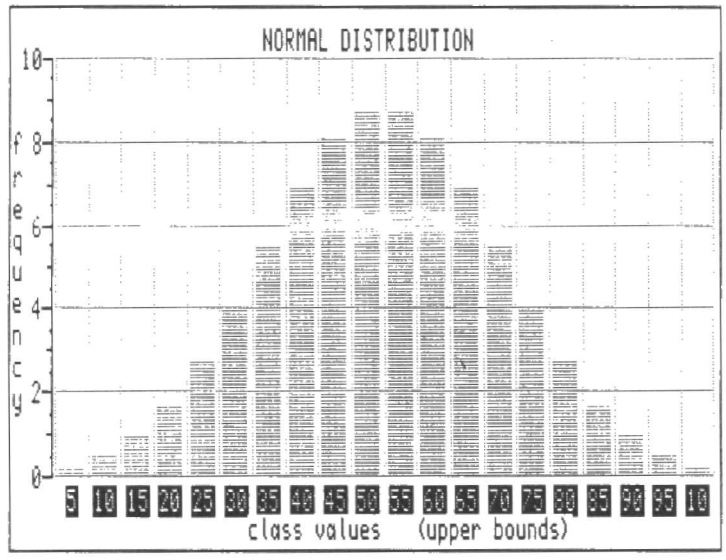
Standard Deviations

In *Archive Answers* this month we are going to have a dip in to the arcane world of statistics, and in particular that shady character, the standard deviation. This is a topic that tends to be restricted to United Nations Statistical digests and university undergraduates, so if you have trouble understanding it, dig out that old jumper with the holes in it and the curry stains, have a few beers, and come back to it after midnight. You'll be amazed at the difference it makes.

If you have dealt with stats before you may remember that the **standard deviation** has a more upright brother called the *normal distribution*, and a black sheep in the family called *Mean*. Mean is the best known, normally called *average*. That name makes statisticians froth at the mouth, but we can't help that. We know about averages, and regularly calculate them for all sorts of things: average price, average goal difference, or whatever. You just add up all the values and divide by the number of items. Easy.

But knowing the average (or mean) is only part of the story. It doesn't tell you how the actual values are distributed *around* the mean. For example, the average of 9 and 11 is 10, but so is the average of 19 and 1. This is where the normal distribution comes in.

In all sorts of realms of study, there are data sets which have this normal distribution. There are also many that don't, but we shan't worry about them. Graph **one** shows the shape of this normal distribution on a frequency chart. Most of the values occur very close to the mean, with the frequency tailing off symmetrically above and below it. People have discovered all sorts of useful things you can deduce if you know your data is normally distributed. It is here that standard deviations come in. But before we come on to that, we'll look at how to make our own normally distributed data.



Creating data

Statisticians are always being accused of creating data to fit the result they want. In our case it is a useful way to get a feel for normally distributed data.

The way to do it is to use the formula in **figure one.** It is the sort of formula that weakens the firmest resolve, full of Greek letters, arithmetic constants, and so on. But fear not. With the help of *Archive*, it can readily be brought under control. One reason for this is that Archive's ancestry can be traced in large part back to Fortran, the computer language devised specifically for formula translation into computer-usable code.

What the formula does is provide a value for 'Y' from any given value of 'x'. On the frequency chart (graph one), x is the class value along the bottom, and Y is the frequency with which that value will occur. So you need to perform the calculation on each value of x you require.

'Pi' and 'e' are both mathematical constants. (Archive has the function 'exp( )' to provide e to the power of a number, and 'Ln( )' to do the opposite.) The other two symbols are bar-x, (the x with a hat on) which denotes the mean; and the lower case sigma (the o with a spout) which denotes the standard deviation. The secret in formula translation is to break the formula down to its simplest elements, calculate these, and then make sure they are combined in the right order.

For any given data set you can choose the *mean* and the *standard deviation*, and calculate any value of Y. The results will all be very small numbers, because the equation is such that the total area under the graph will equal one. To turn it into a bar on a graph it must be multiplied by the width of the bar (the range of each data class). And to give a sensible vertical scale you must decide

figure one - creating normally distributed data

$$Y = \frac{1}{\sigma\sqrt{2\pi}} e^{-z} \quad \text{where } z = \frac{(x-\bar{x})^2}{2\sigma^2}$$

```
proc getnormal
  let standev=18
  let mean=50
  let samplesize=80
  let interval=5
  let min=0
  let max=100
  let class=min+(interval/2)
  let pt1=1/(standev*sqr(pi()*2))
  let pt3=(2*(standev^2))
  kill "normal_dbf": create "normal"
    label$:lessthan:normal: endcreate
  while class<=max
    let pt2=-(abs(class-mean)^2)
    let answer=pt1*exp(-(pt2/pt3))
    let lessthan = class+(interval/2)
    let label$=str$(lessthan,2,0)
    let normal=answer*samplesize*interval
    append
    print lessthan;dec(normal,10,20)
    let class=class+interval
  endwhile
  export "normal";label$,normal
endproc
```

```
HELP  F1 | COMMANDS create look open close | COMMANDS F3
         | delete display back alter find  |
PROMPTS F2 | first insert last next quit   | ESCAPE Press ESC
           | type command & press ENTER (F3 for more)
calculating...
          Grouped frequency statistics on frequency
     6 classes of 1.0
     number of values =              11
     summation of values =           39.5
     arithmetic mean =               3.5909090909091
     standard deviation =            1.8833068443466
         -2S.D. -1S.D.  Mean  +1S.D. +2S.D.
           1.4    2.5    3.6    4.7    5.8
     skewness =                      -0.18083767298585
              (symetrical distribution = 0)
     kurtosis =                      2.4607220789526
              (normal peakiness = 3)

=======================================================

look "potato"
stats;"frequency", yield
```

what is to be the total number of samples in this hypothetical data set, and multiply the result by that as well.

Play around!

The listing for *proc getnormal* in figure one does all of these things, and creates a data file called 'normal_dbf to contain the x and Y values - here named 'lessthan' and 'normal'. It also includes a text field called 'label$', so that the results can be automatically exported. After running *getnormal* you can import the data into *Easel*, to give a good visual idea of the results.

Once you have got it running correctly you can try varying some of the starting parameters. Changing the standard deviation will make the shape wider or thinner; while changing the mean will shift the whole lot left or right. Changing the sample size will alter the vertical scale, and changing the interval will give fewer, wider bars, or more narrow ones. The min and max values decide the parts of the graph to be calculated. In theory you could calculate to infinity in either direction without the Y value actually reaching zero. In practice it will soon be so close that neither you nor Archive can tell the difference.

Deviations

Given that we know what the mean is, what is this standard deviation thing doing?

If you were to draw a vertical line at the mean plus the standard deviation, and another at the mean minus the standard deviation, you will have enclosed 65% of your data samples between the two lines. That is true of all normally distributed data.

And if you were to do the same at the mean plus and minus two times the standard deviation, it will enclose 95% of the samples. So if you know the mean and the standard deviation you can assess how abnormal any one data value is. Alternatively, you can decide what likelihood of occurrence is acceptable, and deduce from that what value you need to be within.

An example may help. You are charged with allocating scarce farmland to *potato*-eating peasants. Each farmer must be able to grow at least four sacks of potatoes each year to avoid starvation. The land is divided into plots, and you have data on how many sacks a plot produced over the last eleven years. Table one shows this data as a frequency table. To calculate the mean, multiply the yields by their frequencies, add them all together and divide by eleven (the sum of the frequencies). If we use mid-points for each yield class we get an average yield of 3.6 sacks per plot. On that basis, would two plots be enough to survive the lean years? If you allocate too much land to each then fewer farmers will benefit.

table one -
frequency of potato yields

Number of sacks harvested per year	Number of years the yield is reached
'lessthan'	'frequency'
1	0
2	1
3	2
4	4
5	3
6	1

The standard deviation of this data is calculated as 1.1, so if we deduct twice this value from the mean it will show the minimum yield in 97.5% of years (the 95% between the two lines, plus the 2.5% above the top one). This gives a yield of 1.4 sacks, so to get four sacks the peasant would need three plots. And that still leaves a little cushion of 0.2 sacks for the awful 2.5% of years we havn't allowed for.

Calculating

How do we calculate the standard deviation? Figure two shows the equation used for grouped frequency data such as this. First the mean (bar-x) must be calculated, using the method already described, and then the main equation. The big M on its side is an upper case 'sigma'. It denotes 'add together for each of the values in the data set'.

This formula requires two passes through the data. Firstly the mean is calculated, and then the mean is used during the standard deviation calculation. There are alternative equations which would do it all in one pass, but they cannot be used by the other two similar looking equations, for *skewness* and *kurtosis*.

These two strangely named statistics are useful in establishing the 'shape' of the data. The skewness measures how symmetrical the shape is. If it is negative the bulge occurs above the mean, and if the skew is positive, most of the values are grouped below the mean, with a long 'tail' of a few larger values. Kurtosis on the other hand measures the shape of the peak. If it is tall (more values than 'normal' clustering in one or two classes) the kurtosis will be greater than three. If it is flattened, with a more dispersed data set it will be less than three. These two values provide a good test of whether data is normally distributed or not. The figures don't depend on the size of the data values. If they stray from their norms by 0.8 or so, the normality of the data is in doubt (the larger the data sample the closer they should be to their norms if the data is to be

figure two - calculating standard deviation from a frequency table.

$$\text{mean} = \bar{x} = \frac{\Sigma fC}{\Sigma f} \qquad \text{standard deviation} = \sigma = \sqrt{\frac{\Sigma f(C-\bar{x})^2}{\Sigma f}}$$

$$\text{skewness} = \sqrt{\frac{\Sigma f(C-\bar{x})^3}{(\Sigma f)\sigma^3}} \qquad \text{kurtosis} = \sqrt{\frac{\Sigma f(C-\bar{x})^4}{(\Sigma f)\sigma^4}}$$

```
proc fstats;n$,c$
  local f,C,sumF,sumFC,sumFC2,sumFC3,sumFC4,ClassInt
  getfield;n$: let field=answer
  getfield;c$: let class=answer
  cls : print "calculating..."
  first : let ClassInt=fieldv(class)
  next : let ClassInt=fieldv(class)-ClassInt
  all : let sumF=sumF+fieldv(field)
    let C=fieldv(class)-(ClassInt/2)
    let sumFC=sumFC+(fieldv(field)*C)
    endall
  let mean=(sumFC/sumF)
  all : let f=fieldv(field)
    let C=fieldv(class)-(ClassInt/2)
    let sumFC2=sumFC2+(f*abs(C-mean)^2)
    if C-mean>0: let sumFC3=sumFC3+(f*((C-mean)^3))
      else : let sumFC3=sumFC3-(f*(abs(C-mean)^3))
      endif
    let sumFC4=sumFC4+(f*(abs(C-mean)^4))
    endall
  let heading$="Grouped frequency statistics on "+n$
  let info$=str(count(),2,0)+" classes of "
+str(ClassInt,0,1)
  let nvalues=sumF: let sum=sumFC
  let standev=sqr(abs(sumFC2/sumF))
  let skewness=sumFC3/(sumF*(standev^3))
  let kurtosis=sumFC4/(sumF*(standev^4))
  spoolon screen :output: spooloff
  endproc
```

```
proc getfield;n$
  let answer=0: while answer<numfld()
    if lower(n$)=lower(fieldn(answer)): return : endif
    let answer=answer+1: endwhile
  print : print "Error : field name not found": stop
  endproc
```

```
proc output
  lprint tab 17;heading$: lprint tab 10;info$
  lprint tab 10;"number of values ="; tab 40;nvalues
  lprint tab 10;"sumation of values = "; tab 40;sum
  lprint tab 10;"arithmetic mean ="; tab 40;mean
  lprint tab 10;"standard deviation = "; tab 40;standev
  lprint tab 15;"-2S.D. -1S.D.  Mean  +1S.D. +2S.D."
  lprint tab 13;: let sd=-2*standev
  while sd<3*standev: lprint dec(mean+sd,1,7);
    let sd=sd+standev: endwhile : lprint
  lprint tab 10;"skewness ="; tab 40;skewness
  lprint tab 20;"(symetrical distribution = 0)"
  lprint tab 10;"kurtosis ="; tab 40;kurtosis
  lprint tab 20;"(normal peakiness = 3)"
  endproc
```

considered normal).

The three procedures listed in figure two enable calculation of these formulae. To use them you need an open data file, and call *'fstats'* (standing for frequency statistics) from the Archive prompt. Two field names are needed as parameters. The first is the field holding the frequency data; the second should contain the upper limits of each class. Both must be in inverted commas, separated by a comma. Fstats can be used by any data file containing such information. The results are initially sent to the screen (as shown in the screen dump), but can also be sent to the printer afterwards by entering 'output' at the Archive prompt.

If you create a data file with fields for 'lessthan' and 'frequency' and insert the data in table one you can test the procedure by entering:

fstats;'frequency','lessthan'

You should get results like those in the screen dump. You can also use fstats to test the data you created using getnormal. In this case the mean and standard deviation should end up close to your starting values, with a skewness near zero, and a kurtosis near three. If these are more than a tiny fraction out (due to rounding at 14 significant figures!) you should re-check your typing.

Raw data

Data doesn't always arrive in frequency table form! More often than not it will start as a large number of individual numbers. To analyse such data we can either group it together into frequency form, or perform the calculations directly. We shall look at ways to do both of these, starting with the latter. Figure three shows a set of formulae similar to those in figure two. However, instead of C, the class value, we have x, the data itself. And there is no figure for f, as each value represents just itself, instead of a group of values represented by each frequency entry.

table two - component repair times

Component code	Repair/replacement time (minutes)	
'part$'	'designA'	'designB'
A	25	33
B	8	12
C	34	24
D	12	7
E	21	18
F	45	16
G	2	23
H	27	20
I	16	26
J	14	27

figure three - standard deviation using raw data

$$\text{mean} = \bar{x} = \frac{\sum x}{n} \qquad \text{standard deviation} = \sigma = \sqrt{\frac{\sum (x-\bar{x})^2}{n}}$$

$$\text{skewness} = \sqrt{\frac{\sum (x-\bar{x})^3}{n\sigma^3}} \qquad \text{kurtosis} = \sqrt{\frac{\sum (x-\bar{x})^4}{n\sigma^4}}$$

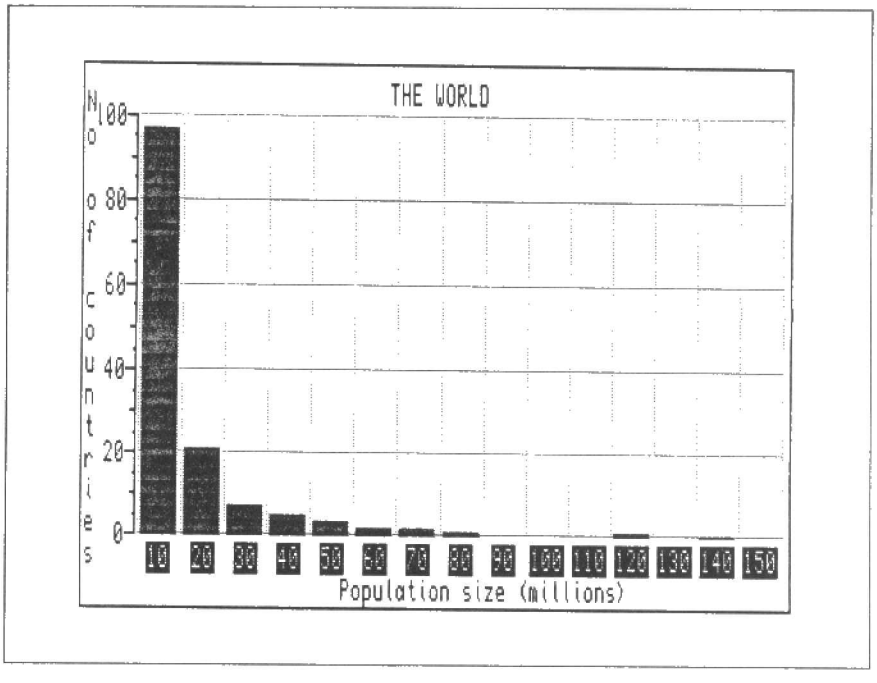
```
proc stats;n$
  local sumX,sumX2,sumX3,sumX4,min,max
  get.field:n$: let field=answer
  clw : print "calculating ..."
  let min=9.E+37: let max=0
  all : let xbar=fieldv(field)
    let sumX=sumX+xbar
    if min>xbar: let min=xbar: endif
    if max<xbar: let max=xbar: endif
    endnl
  let nvalues=count()
  let mean=sumX/nvalues
  all : let xbar=fieldv(field)-mean
    let sumX2=sumX2+abs(xbar)^2
    if xbar>0: let sumX3=sumX3+(xbar^3)
      else : let sumX3=sumX3-(abs(xbar)^3)
      endif
    let sumX4=sumX4+(abs(xbar)^4)
    endall
  let heading$="STATISTICS ON "+upper(n$)
  let info$="range : "+str(min,2,0)+" to "
+str(max,2,0)
  let sum=sumX
  let standev=sqr(sumX2/nvalues)
  let skewness=sumX3/(nvalues*(standev^3))
  let kurtosis=sumX4/(nvalues*(standev^4))
  &xecute screen :output: spooloff
  endproc
```

The listing contains just one procedure, *stats*, which uses two of the procedures already typed in. It has the same output option as before. This time only one parameter is needed - the name of the numeric field to be analysed. **Table two** provides another example, with two sets of figures. The scenario is a high-powered electronics company, building control equipment for critical applications. The customer's order specifies a mean repair time of 20 minutes, but also that 97% of faults must be repairable in 38 minutes. No doubt they have their reasons. (The nuclear core melts if the electronics are down for 40 minutes, say.) Your engineers produce two rival designs, design A being substantially cheaper to make. Will it fit the bill? Table two is the repair time of all the components. If you enter the figures into a database and run each design through stats you should find that both just about meet the 20 minute average. However the 97% figure (mean plus two standard deviations) for design A is nearly 45 minutes, by which time things are getting too hot to handle. Design B on the other hand fits it in at 35 minutes, leaving three minutes to find that lost screw.

Hidden inside

You can try *stats* on some other data you may have. The 'Gazet' database supplied with Archive provides data on the population of the countries of the world, in the field 'pop'. What happens if you use this? The mean looks good, but the other data is haywire. The frequency graph of the data, shown in **graph two** may explain why. There are lots of very small countries, and just a few huge ones (some well off the scale of this graph). However, believe it or not, there is a normal distribution hidden somewhere inside these figures. They just need 'transforming' to bring it out. The population distribution, along with all sorts of unlikely bedfellows, has a 'log-normal' distribution. *Proc transform* in **figure four** creates a new database containing the population figures expressed as natural logarithms. If you perform a stats analysis on the transformed data, low and behold the figures indicate a normal distribution.

You need to be aware that they are now log results, and to make sense they need anti-logging with the exp( ) function. When to anti-log is of prime importance here. In particular if adding or subtracting standard deviations the logged figures are used, and the result anti-logged. For example, you are charged with dividing up an ex-communist state into its



constituent ethnic groups, but the UN has decided the units must not be smaller than 80% of the world's countries. You will find that the log-mean minus the log-deviation equals 13.8. Anti-logging this reveals that provided you ensure a million people per country you will be above 82%. (65% between the first standard deviations, plus 17.5% above that.)

Frequency form

We can further verify that the data is normal by graphing the result. To do this the procedure 'group', in **figure four,** will produce yet another file, this time grouping the log-population data into frequency form. As with all 'derived' data the file can be regarded as temporary. To save file space you can safely delete everything except the raw data (in this case gazet_dbf) and the program file. The other files can all be easily regenerated at any time.

It would be a fairly simple matter to adapt *proc group* to do the same for any raw data. The lines you may need to customise are marked with numbered arrows. Number 1 is simply the name of the source file. At 2 you must decide an appropriate class interval, and at 3 you need to enter the minimum and maximum classes. At 4 you may wish to alter the text labels, and at 5 you must enter the name of the field to use in the source file.

Proc group produces both a data file, and an export file, this time called summary_exp. Importing this to Easel should produce something like graph three, here shown with an additional *completely normal curve*, generated by getnormal using the same mean and standard deviation. For all the usefulness of the various figures we have produced, there is probably no substitute for the old eyeball method for getting to grips with a set of numbers.

TO DO LIST UPDATE

Archive enthusiasts of long-standing will remember the *Archive Power* series a couple of years ago, which built up a set of "desktop" tools. The final instalment, in October 1990, was for a To Do List and Diary. The request has come across the North Sea for a modification. Instead of just showing one day's diary entries they wanted to show a week or more of appointments on one screen. I present here the modified version of proc LISTDRAW. The changes are surprisingly slight, for such a major specification change, due to the power and flexibility of *Archive*.

There are two other changes needed in addition to the emboldened parts of LISTDRAW. You need to add a line to proc LISTLINE. Immediately below the line that says 'if count()>0' you should add the line:

if DAYS and DAYS<>days(CURRENT$): let PRIORITY$=DATE$(6 to): endif

And at the end of the 7th line of proc LIST, you should change it to read:

let colour=3+(DATE$=CURRENT$)

And that is all there is to it. You can choose how many days ahead you want to include on your screen, by editing Listdraw, and changing the last number of the first line. As shown here it will list entries for six days ahead of the current date. You could add an option to set this value from within the program with a bit more effort.

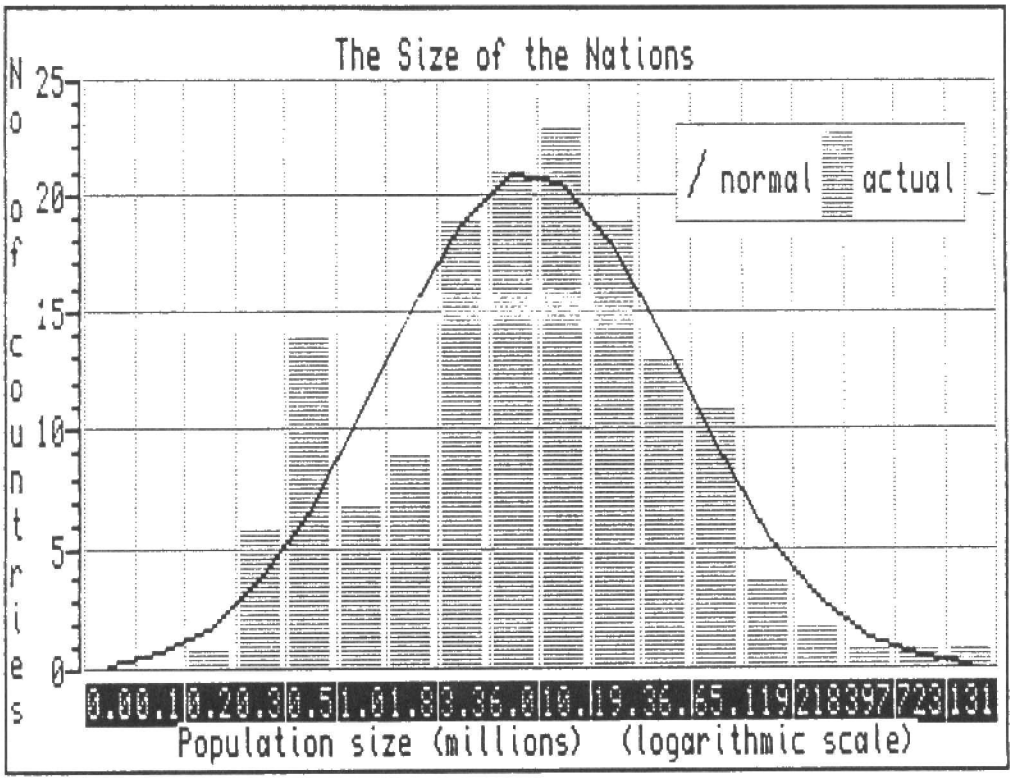
figure four - transforming and grouping data

```
proc transform
  create "popln" logical "ln"
    pop: endcreate
  look "gazet" logical "source"
  all "source"
    let ln.pop=ln(source.pop*1000000)
    append "ln": endall
  close "ln": close "source"
  print "finished."
  endproc
```

```
proc group
  look "popln" logical "source":rem      <=== 1
  kill "summary_dbf": print "creating..."
  create "summary" logical "sum"
    label$:lessthan:frequency
    endcreate
  local interval: let interval=0.6:rem   <=== 2
  let count=11.4: while count<=22.2:rem   <== 3
    let label$=str(count,0,1):rem         <=== 4
    let lessthan=count
    append : let count=count+interval
    endwhile : order lessthan;a
  print "calculating..."
  all "source"
    use "sum": locate (source.pop):rem   <=== 5
    let frequency=frequency+1: update "sum"
    print recnum("source")
    use "source": endall
  close "source": print "exporting..."
  use "sum": export "summary";label$,frequency
  endproc
```

Modified procedure for To Do List. Altered and inserted code is shown in bold.

```
proc LISTDRAW
  local FROM,UPTO: let FROM=days(CURRENT$): let UPTO=FROM+6
  DATEFORMAT;CURRENT$: print HEADING$
  if CURRENT$=date(0): let ANSWER$=ANSWER$+" **TODAY***": endif
  print at 4,0; ink 0; paper 4;"  ";ANSWER$; tab 75;LISTWINDOW$
  cls : if REDRAW=2 and count(): reset : order DAYS;A,PRIORITY$;A
    if CURRENT$<=date(0): select DAYS<=UPTO
      else : select DAYS>=FROM and DAYS<=UPTO: endif : endif
  last : while recnum()>0
    LISTLINE;0: print ink 3+(DATE$=CURRENT$);ANSWER$;chr(22)+chr(1);
    back : endwhile
  LISTLINE;0: print ANSWER$;
  endproc
```



Simon Goodwin brings Qdos into the Laser Zone

DIY TOOLKIT

HPDUMP is a fast, adjustable screen image printer for modern page printers. It supports four different print formats, ranging from roughly A5, down to A7 - about 90 x 65 mm - with normal or inverted brightness and half tone shading. The project illustrates the difference between Epson and HP printer controls, and shows a quick, concise way to unscramble QL display memory.

HPDUMP is of interest even if you do not own a page printer, because it shows how one source file can generate code for an executable task, a resident procedure, or a CALL routine, rather like Psion's GPRINT_PRT.

This month's SuperBasic loader generates the resident version of HPDUMP. Next month I shall present code for a version that suits EXEC and CALL. Both code files are based on the same source. This month Listing One shows the parameter handling and initialisation routines. Next issue I shall complete the listing with assembly code to read the QL screen and generate a corresponding HP image.

New drivers

The QL user is well supported by dot-matrix printer drivers, from the original Epson FX-80 driver supplied with Easel, via Tasman's Tasprint extension, through to 24-bit graphics drivers from Dilwyn Jones and Digital Precision. Qram, Gold Card and Trump Card have a range of graphics drivers built in, but again these are aimed at cheap, slow and noisy matrix printers.

The latest inkjet, LED and laser printers work rather differently, so they need new graphics drivers. A few will emulate Epson codes, so they can work with dot-matrix drivers, but the different dot shape means that images get squashed or stretched, and you lose the high resolution and extra modes of the new printers.

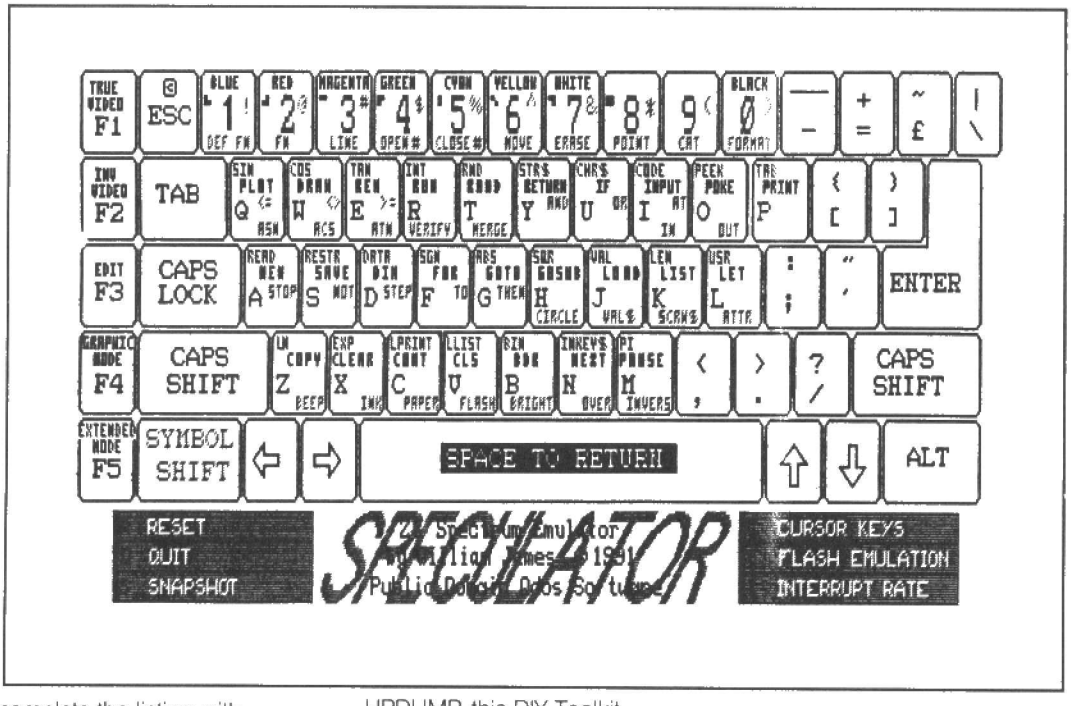
Top notch page printers recognise Postscript, a symbolic interpreted graphics language, but this is greedy for memory and processor power, so it is only supported by the most expensive printers. But almost all page printers recognise Hewlett Packard's 'PCL' Printer Control Language, so that is the new standard addressed by HPDUMP, this DIY Toolkit project.

HPDUMP has been carefully tested on my £300 DeskJet 500 printer, and Dr Bill Fuggle's upmarket LaserJet 2. It should suit all page printers that accept PCL codes. It can 'dump' Mode8 as well as Mode4 screens, rendering colours as shades of grey. You may improve results for images with low contrast by using RECOL before printing.

The easy way

HPDUMP is central to the latest DIY Toolkit volume, devoted to HP Print Control Language. Volume K includes the task, CALL and command versions of HPDUMP, full source code, configuration data for Quill and Xchange, a TRA code translator for all Qdos variants except AH and JM, and dozens of SuperBasic extensions to select founts and print styles.

24 DIY Toolkit volumes are now available on disk and microdrive, at a new flat rate of £3 each on 3.5 or 5.25 inch Qdos disks, or £4 per volume on cartridge. Order two or more volumes to receive printed documentation at no extra charge. To obtain volumes, or further information, write to the new ***DIY librarian Dr. Bill Fuggle at DIY Toolkit, 86 Lordswood Road, Harborne, Birmingham B17 9BY, UK.***



The listings

Listing One is the first half of the assembly code for HPDUMP. The file uses conditional assembly, so it can generate code for a task or a resident procedure, depending on the value assigned to EXTENSION. This complicates the listing but means that there is no need to update two files whenever a change is made in the large common part of the code.

Code bracketed by IFEQ EXTENSION and ENDC is only generated if EXTENSION is zero - IFEQ means 'if this symbol is equal to zero'. Similarly, IFNE stands for 'if not equal to zero'. Code thereafter is only assembled if EXTENSION is true, usually signified by a '1' value, indicating resident code.

Most QL Assemblers support conditional code, including HiSoft's Devpac, Metacomco's ASM and GST's Macro Assembler. If yours does not, you must decide whether you want the task or resident version, and enter only those lines marked IFEQ (for the task) or IFNE (for the extension).

If all else fails, pick the task code and use Task Commander, from DIY Toolkit Volume F, to convert it into a keyword - but this will use more code and slightly different syntax, as the task version expects to find the "!" indicator inside the parameter string.

Assembly Code

The first block of conditional code handles task parameters. DATA command sets the task data-space - 640 bytes for conversion tables, plus a few for the task's stack. The code labelled MULTIDUMP picks parameters off the stack. This is the first DIY extension implemented as a task; October 1991's Task Commander tacked the problem from the opposite end, setting up parameters to call a task.

When the task starts there are at least two words on the stack,

at the end of the task's data space. The first word indicates how many channels are passed to the task, followed by zero or more channel ID's of four bytes each. The next word is the length of the parameter string, which may range from zero to 32767 bytes. By default EXEC stores two zero words, but Task Commander and Toolkit commands like EX, EXECUTE and QX pass parameters by storing larger values.

The alternative code for the HPDUMP command is prefixed by IFNE EXTENSION. This is much more conventional DIY Toolkit fare, starting with a jump to BP.INIT to initialise the command, and using other rom vectors to fetch integer or string parameters. The command and the CALL variant also need 640 bytes for data tables, which they request from the Common Heap with MT.ALCHP. D7 is used as a flag so that the code knows whether to close the channel or de-allocate memory when it has finished.

Once the channel ID and table space are ready HPDUMP calls MT.INF to find the system variables, and checks bit 7 of SV.MCSTAT, the ZX-8301 status, to see if Minerva's second screen is in use. It starts out by pointing A4 at the second screen, moving it back by 32K for screen 1. This is a tad more efficient than assuming screen 1, as 68000 family processors use 'two's complement' arithmetic, treating -32768 as a word and +32768 as a long word value!

This esoteric observation concludes the discussion of Listing One, which will be completed next issue with code to talk to the printer. Further notes appear in the DIY disk files and my next column.

Another easy way

Listing Two is a SuperBasic loader that generates the code for the HPDUMP extension. The first part, up to line 470, is Marcus Jeffrey's hex loader, as used in each DIY Toolkit project. This program reads and checks the hex data from line 580 onwards. If all is well it asks you to nominate a name for the code file.

Once the file FLP1_HPDUMP_CODE (or whatever you choose) has been created, type these commands to link the HPDUMP extension to SuperBasic:

```
X=RESPR(484)LBYTES
FLP1_HPDUMP_CODE,XCALL X
```

Once you have done this, the command HPDUMP will send an A5 screen dump to the SER port.

The code accepts channel and string parameters, so you can over-ride the default device without re-assembling the code. This is particularly useful if you use a printer buffer, such as Miracle's PRT device, or a parallel port like PAR on the Thor, Amiga or SuperQboard.

A full screen dump generates 85-100K of data. HPDUMP needs little more than a second to work this out from the QL screen, but it takes a couple of minutes to squirt it out through Sinclair's slow serial ports. HPDUMP goes twice as fast if you have a parallel port, capable of sending eight bits at a time to the printer. It should work faster still if your printer out-performs my Amiga Qdos/DeskJet 500 combination.

Resolution

Typical page printers have a maximum resolution of 300 dots per inch, known as 300 DPI. HPDUMP uses two dots horizontally and two or three vertically for each QL pixel, so that colours can be distinguished by shading.

The default printout uses four dots by four or six, without generating any more data, by selecting 'double size' 150 DPI before sending the dots. You can over-ride this by replacing the 150 DPI message "*t150R" at the end of the dump routine with its 300 DPI equivalent "*t300R", to get the small dump. You may need to reset the printer before the new size takes effect. I shall reveal more PCL codes next month.

The Mode4 screen is 512 pixels wide and 256 deep, so you might expect HPDUMP to use three dots for every pixel in the vertical direction, as most screens have a 4:3 aspect ratio, like a TV - in other words, they are one third wider than they are tall.

Last minute bodges mean

```
* QL World DIY Toolkit, HPDUMP for Qdos screens
* Version 2.18, Copyright Simon N Goodwin March/April 1993
*
extension   equ     0                   1 = resident, 0 = transient
*
            ifeq    extension
*
            data    660                 Task buffer and stack space
*
task_start  bra.s   multidump           6 bytes for initialisation
            dc.b    '2.18'              Version number fills four
            dc.w    $4AFB               Task header
            dc.w    6                   Name length
            dc.b    'HPDUMP'            Task name
*
multidump   lea.l   task_start,a0       Where am I ?
            cmpa.l  a0,a6               If A6 is here, I'm a task
            bne.s   calldump
            moveq   #1,d7               Flag task invocation
            move.w  (a7)+,d0            Are any channels supplied?
            beq.s   string_par
            moveq   #-15,d3             Preset error code ERR.BP
            subq.w  #1,d0
            bne.s   remove_task         Only one channel, please!
            moveq   #-1,d7              Flag: channel passed to task
            moves.l (a7)+,a0            Fetch the channel ID
*
string_par  moveq   #0,d5               Presume not inverse
            move.w  (a7),d0             Any parameter string?
            ble.s   default_ch
strung      cmpi.b  #'!',1(a7,d0.w)     Check the end
            bne.s   not_inverse
            moveq   #-1,d5              Set inverse mask
            subq.w  #1,(a7)             Correct length
            beq.s   default_ch          No name left
not_inverse tst.l   d7                  Do we have a channel yet?
            bmi.s   channel_set         If so, no need to OPEN one
            moves.l a7,a0               Point A0 at parameter$
*
* N.B. This only works for tasks; A7 must be static (i.e. not BASIC)
*
            bra.s   chosen_ch
*
default_ch  tst.l   d7                  Is ID known yet?
            bmi.s   got_channel
            lea.l   printname,a0        Point at default name
chosen_ch   moveq   #-1,d1              Own channel
            moveq   #2,d3               New exclusive device
            moveq   #1,d0               IO.OPEN
            trap    #2
            move.l  d0,d3               Copy and test error code
            bne.s   remove_task         How disappointing
            moveq   #1,d0
            add.w   (a7)+,d0            Round up length
            bclr    #0,d0               Keep result even
            add.l   d0,a7               Tidy name from stack
*
got_channel lea.l   0(a6,a4),a5         Point A5 at task data area
            move.l  a0,d6               Save channel ID
            bsr.s   set_buffer
            move.l  d0,d3               Check error code
* D3 is the error code reply for the calling task
*
remove_task moveq   #-1,d1
            moveq   #5,d0               MT.FRJOB
            trap    #1                  There's no return from here!
*
calldump    moveq   #-1,d5              Inverse mask (true brightness)
            tst.l   d1                  Is parameter non-zero
            bne.s   open_out
            moveq   #0,d5               Non-inverting mask
*
            endc    call/task
*
* The BP.INIT code is only needed if this is an extension file
*
            ifne    extension
*
start       lea.l   define,a1
            move.w  $110.w,a0           BP.INIT vector
            jmp     (a0)
*
* If HPDUMP has a "!" separator the INVERSE dump is selected.
* # signals a SuperBasic channel number, otherwise a device name.
*
hpdump      moveq   #0,d5               Non-inverting maybe?
            cmpa.l  a3,a5               Are there any parameters?
            beq.s   open_out
            moveq   #%111000,d0         Mask out separator
            and.b   1(a3,a6.l),d0
            cmp.b   #%100000,d0         Is it an exclamation mark?
            bne.s   no_shriek
            moveq   #-1,d5              Inverting mask
no_shriek   tst.b   1(a3,a6.l)          Is there a # prefix?
            bmi.s   hash_prefix
*
not_channel moves.w $116.w,a0           CA.GTSTR vector
            jsr     (a0)
            bne.s   bad_return
            subq.w  #1,d3               One parameter?
            bne.s   bad_param
            moves.l a1,a0               Point A0 at the string
            trap    #4                  String is A6 relative
            bra.s   open_param          Try to open it
*
hash_prefix moves.w $112.w,a0           CA.GTINT vector
            jsr     (a0)
            bne.s   bad_return
            subq.w  #1,d3               One parameter?
            bne.s   bad_param
            moves.w 0(a1,a6.l),d0       Fetch parameter word
            bmi.s   bad_param
            mulu    #40,d0              Scale for SV.CH entries
            add.l   48(a6),d0           Add SV.CHBAS offset
            cmp.l   52(a6),d0           Check against SV.CHP
            bge.s   bad_channel
            move.l  0(a6,d0.l),d0       Fetch channel ID from table
            bpl.s   channel_ok
*
bad_channel moveq   #-6,d0              Channel not open
            rts
*
bad_param   moveq   #-15,d0
bad_return  rts
*
channel_ok  moves.l d0,a0               Move ID to usual register
            bra.s   channel_set
            endc    extension
*
* Open the output channel
*
open_out    lea.l   printname,a0
open_param  moveq   #0,d7               Indicate transient type
            moveq   #-1,d1              Own channel
            moveq   #0,d3               Old exclusive device
            moveq   #1,d0               IO.OPEN
            trap    #2
            tst.l   d0                  Test error code
            bne     bad_exit
*
channel_set move.l  a0,d6               Save channel ID
allocate    move.l  #128+256*2,d1       640 bytes of buffer
            moveq   #-1,d2              Owned by this job
            moveq   #24,d0
            trap    #1                  MT.ALCHP
            move.l  d0,d4               Copy and test error code
            bne     tidy_up             Out of memory!
            moves.l a0,a5               A5 -> Line buffer base
set_buffer  lea.l   128(a5),a2          A2 -> Lookup table
*
* Locate the screen
*
            move.l  #131072+32768,a4    Assume screen 1 for now
            moveq   #0,d0
            trap    #1                  MT.INF
            move.b  52(a0),d0           Test SV.MCSTAT
            bmi.s   screen1             Top bit means screen 1
            lea.l   -32768(a4),a4       Wind back to screen 0
screen1     moves.l d6,a0               Restore channel ID
*
*
*
*
*
*
go
```

```
DIY Toolkit - HPDUMP hex loader - LISTING 2

100 REMark Sinclair QL World HEX LOADER v 3a
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
130 CLS: RESTORE : READ space: start=RESPR(space)
140 PRINT "Loading Hex..." : HEX_LOAD start
150 INPUT "Save to file...";f$
160 SBYTES f$,start,byte : STOP
170 :
180 DEFine FuNction DECIMAL(x)
190 RETurn CODE(h$(x))-48-7*(h$(x)>"9")
200 END DEFine DECIMAL
210 :
220 DEFine PROCedure HEX_LOAD(start)
230 byte = 0 : checksum = 0
240 REPeat load_hex_digits
250    READ h$
260    IF h$="*" : EXIT load_hex_digits
270    IF LEN(h$) MOD 2
280       PRINT"Odd number of hex digits in: ";h$
290       STOP
300    END IF
310    FOR b = 1 TO LEN(h$) STEP 2
320       hb = DECIMAL(b) : lb = DECIMAL(b+1)
330       IF hb<0 OR hb>15 OR lb<0 OR lb>15
340          PRINT"Illegal hex digit in: ";h$ : STOP
350       END IF
360       POKE start+byte,16*hb+lb
370       checksum = checksum + 16*hb + lb
380       byte = byte + 1
390    END FOR b
400 END REPeat load_hex_digits
410 READ check
420 IF check <> checksum
430   PRINT "Checksum incorrect. Recheck data.":STOP
440 END IF
450 PRINT "Checksum correct, data entered at: ";start
460 END DEFine HEX_LOAD
470 :
580 REMark Space requirements for the machine code
590 DATA 484
600 :
610 REMark Machine code data
620 DATA "43FA01D030780110","4ED07A00BBCB6758"
630 DATA "7038C033E801B03C","002066027AFF4A33"
640 DATA "E8016B1230780116","4E90663653436630"
650 DATA "20494E4460363078","01124E9066245343"
660 DATA "661E3031E8006B18","C0FC0028D0AE0030"
670 DATA "B0AE00346C062036","08006A0870FA4E75"
680 DATA "70F14E7520406014","41FA01467E0072FF"
690 DATA "760070014E424A80","660001042C08223C"
700 DATA "0000028074FF7018","4E412800660000EA"
710 DATA "2A4845ED00B0287C","0002800070004E41"
720 DATA "102800346B0449EC","8000204676FF43FA"
730 DATA "01103419610000DA","264A43FA00E4700F"
740 DATA "720F3400E94AD441","D44218310000E14C"
750 DATA "D831100037842000","51C9FFE851C8FFE2"
760 DATA "3C3C00FF703F264D","7200121CD2413432"
770 DATA "1000D4427200121C","D24184721000BB42"
780 DATA "36C251C8FFE46178","49ECFF80703F264D"
790 DATA "72001214D2413432","10007200121CC21C"
800 DATA "D24132321000D241","8441BB4236C251C8"
810 DATA "FFE0614C08060000","662A49ECFF80703F"
820 DATA "264D7200121CD241","343210007200121C"
830 DATA "D24132321000D241","8441BB4236C251C8"
840 DATA "FFE2611C51CEFF7E","78004A876B0E6608"
850 DATA "C14D70194E41C14D","70024E4220044E75"
860 DATA "43FA003434196110","660A224D343C0080"
870 DATA "6106670A588F60D2","70074E4328004E75"
880 DATA "0001040510111415","4041444550515455"
890 DATA "0004534552310007","1B2A623132385700"
900 DATA "000F1B2A74313530","521B2A7231303234"
910 DATA "53000001FE360648","5044554D50000000"
920 DATA "00000000","*",38996
```

on one A4 page, with a little room left for captions. My DeskJet saves time and ink if I select 'draft' output, but this skips some dots so green and white pixels are almost indistinguishable in a small draft printout.

Printer paper is most often white, and screen backgrounds are usually black, so HPDUMP defaults to an 'inverse' image, with black and white transposed, substituting light grey for red and dark grey for green. If you have plenty of ink or toner you can get true grey scale by adding an exclamation mark after the command, eg:

HPDUMP !HPDUMP #3!HPDUMP "PARF"!

Note that solid ink in draft printouts can look rather streaky. Narrow white lines surrounded by big black areas may disappear unless you select draft before generating a small dump at 300 DPI, particularly with absorbent paper on an inkjet printer. This should not happen if you select the larger 150 DPI image or use a laser printer.

The full image distinguishes all QL colours by the use of shading. HPDUMP can also be used to generate full-colour printouts if you have the necessary coloured ink or toner cartridges. I shall return to this topic next month.

Coming soon

My next column will reveal the code that converts a QL screen into printer control codes, and the multi-tasking version, HPDUMP_TASK, which can also be used as a replacement for Psion's GPRINT_PRT in programs that CALL that code. As ever I'm keen to hear your experiences and suggestions about DIY Toolkit projects; please write to me, care of QL World.

that the QL screen is stretched horizontally, which is why its leftmost columns disappear into the margin of many TV screens. HPDUMP needs to compensate for this to preserve the shape of printed screens, so the vertical scale factor averages 2.5 dots per pixel.

This means that circles generated by Easel and SuperBasic still appear round when printed. Next issue I shall explain how to control the aspect ratio; the 4:3 variant gives stretched 'circles', slightly larger dumps and neater grey tones. The accompanying example shows the Speculator help screen, at three dots per pixel, while the sample screen in this month's SuperBasic in Action column averages 2.5 dots per pixel.

HPDUMP can take an optional channel or a string parameter. If a channel is supplied HPDUMP sends output to the channel, leaving it open for further use when it has finished. Thus you can build up a file of text and graphics on any drive:

```
OPEN_NEW #3,RAM1_REPORT
LBYTES FLP1_PROFIT_SCREEN,131072
HPDUMP #3
PRINT #3,\\"Profits for 1991-92"\\
LBYTES FLP1_FUTURE_SCREEN,131072
HPDUMP #3
PRINT #3,\\"Projection for 1993"
CLOSE #3
```

This sequence generates a file RAM1_REPORT of about 200K, incorporating two screens with text captions. You can print this file any number of times by copying it to your printer device. The screens are 32K ram images, generated by saving the image from display memory. You can make such files with this SuperBasic command:

```
SBYTES FLP1_FUTURE_SCREEN,131072,32768
```

Alternatively, use Dilwyn Jones' multi-tasking Screen Snatcher utility, or generate the screen file with a graphics program like Easel, Eye-Q or The Painter.

Economies

By default HPDUMP sends an half-page image (just within A5) to the printer. The smaller 300 DPI image lets you fit eight

Text'n'Graphix

Henry Orlowski mixes text and pictures with a low-price importing program.

INFORMATION
Program: Text'n'Graphix
Price: £20 (demo version £2)
Supplier: Dilwyn Jones Computing, 41 Bro Emrys, Tal-Y-Bont, Bangor, Gwynedd LL57 3YT.
Requirements: 3.5in disk driver, Epson compatible printer.

All QL users have a wordprocessing package even if it's only *Quill*. Many use a graphics program even if it's only *Easel*, or perhaps they are familiar with the QL's graphics capabilities and screen save facilities. Some may quite like a relatively inexpensive method of putting their graphics within their text in a single printout without having to go to full blown desktop publishing (DTP) and having to learn totally new processes from scratch.

Text'n'Graphix (TnG) is a designed to control and enhance your printed text output and to mix graphics into it. The final effects can approach that of DTP. Unlike a self-contained DTP package, TnG relies on separate textual and graphical input from different programs which it can then 'grab' and combine into a single printout, although it will not actually display your mixed text and graphics output on screen.

Let's take first the graphics part of the program, as this is probably the major attraction to the potential user.

Grabbing

This is not a graphics program. It will not in itself allow you to create pictures. What it will do is allow you to either load screens saved previously, or to multitask with other graphics programs from which you can save a display. If you're loading a saved screen, remember that most graphics packages have two screen-save alternatives. The default one is usually the compressed or contracted one. These use much less disk space, but most compression routines are different, which means there is little or no compatibility between different screen files. TnG therefore does not load these (unless it's one of its own). If you do try to load a compressed file, all you will get is a top portion of the screen filled with a random colour pixel arrangement.

The screen save method you should opt for is the screen byte format and most programs give you this option. This saves the screen in the same way as the SBYTES command or as printing, say, an Easel screen dump to a file.

TnG also handles any differences between the mode the screen was saved in and the mode you are currently running in. If you get the odd strange effect on screen, compensation is made so that any final printout is correct.

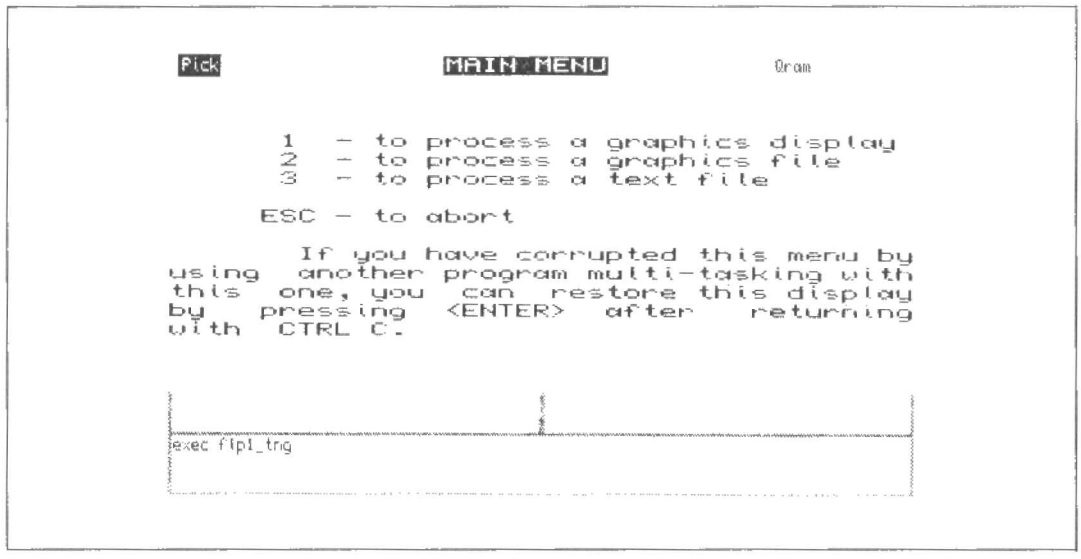
If you haven't got a saved screen to load you may choose to multitask with a separate graphics program using Ctrl-C to switch between them. Switching back to TnG once you have got your display ready allows you to capture the display or part of it, in the same way as you can with a saved screen. TnG gives you a movable, resizeable XOR'd block to enable you to define the portion of the screen for processing. You use Ctrl or Alt plus the arrow keys to move the block around the screen and to change its size. This takes a little getting used to, but you soon get the hang of it.

Options

Once you have decided on the part of the screen you want you can do three things to it. Firstly, you can just dump it to the printer. Secondly, you can store it for later use as a TnG contracted file. The third option is the one you will have selected if you wish to use the graphics within a text document. This option not only stores it as a contracted file but creates a special insert file which can be subsequently placed within a document. A fourth option allows you to do all three.

If you have stored a contracted TnG file you can call it up to make any further modifications to it. Typical modifications include altering the height and/or width of the display, or storing several smaller sections from the same display. Once you have completed the task you can again either dump to printer, create a new contracted file(s), or create a special file for insertion into a document.

That covers the graphics part. Now let's look at the text part. Like the graphics, you don't use TnG to create the text part. What you have to do is load up your favourite wordprocessor or editor. You key in your text and intersperse it with the TnG control codes. These control codes represent the various printer functions that allow you to achieve special effects within your documents. They are vital because they are recognisable to the TnG printer driver which will then implement the function on your printer. You can put in as many control codes as you can fit in and are practicable, and you can change the effect you give to your text even on a line by line basis.



In use

What are these functions that the control codes can implement? I'm not going to list all 108 of them, but I will give an example of how they can be used. Take your document. You want the title to be bold and enlarged. Just put in the appropriate control codes for that. Then you want to revert to normal size text but NLQ, so put in those codes. Then there are some sections that you want to highlight by making them italic. Easy. Then some notes sections that you want in condensed print with reduced margins. Do it. TnG will happily handle it for you.

Don't be put off by the prospect of having to key in these control codes. After a short while they become second

SIMPLE TnG EXAMPLE

Just a simple example to see if we can mix text with pix.
Put the picture in here:

nature. The manual lists all those that are implemented, about 108 of them, most of which are likely to be available on your printer if it is Epson-compatible. If it is not or it varies in some way from the standard then it is a relatively straightforward operation to set up the control codes that your printer will recognise. (Note however that Epson compatibility is a pre-requisite for the graphics routines to work.)

Most of the control codes are merely sequences of 3, 4 or 5 characters initialled by Ctrl-Esc or Ctrl-Shift keypress combinations. They can with thought provide a powerful level of control over your printed output with effects that can be quite impressive.

Picture it

How about getting the graphics into the text? Remember the special insert file of our graphics screen dump we mentioned earlier as one of the options when processing your graphics display. This is the file that we have to input into our document. This depends on the wordprocessor you are running. In the case of Quill you use the Files menu and then the Import option, which will place the file at the cursor position. When importing, the insert file will automatically shift down any existing text the appropriate number of lines to allow the graphics space within the printout.

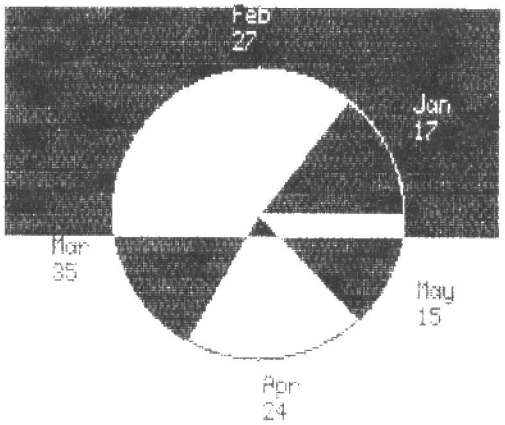
There are then just two things to remember. Firstly, put your text control codes back in at the end of the insert file space. This is because the graphics routines remove them all so they have to be re-set. Secondly, save your final version as an Ascii plain text file. If it's Quill you do this by printing to a _lis file.

That done, you exit Quill and load up TnG or switch back to it if you are multitasking. From the main menu choose the option to process a text file. This then gives you two further options. You can either choose to print out your text and graphics creation there and then or you can save the whole thing as a special file which can be used without having to go through TnG at a later time. All you then would have to do to print it is to use SuperBasic to Copy the file to your printer port, ser1 or par or whatever.

You do need to spend some time on TnG to get the best out of it. The author, Alan Bridewell, has provided a whole range of aids and features to enable you to grasp it all as easily and quickly as possible.

Examples

The manual is comprehensive, instructive and logically presented, and leads you through a number of examples using some files included on the disk. A number of sample graphics screens are also provided to help you practise loading and processing displays. There is even a simple but effective graphics creation program to allow you to practise setting up your own screen displays while multitasking with TnG.



Although quite complex in the variety of operations you have to perform, TnG is driven by simple menu selection and on-screen instruction and information.

You can opt for sound (progress beeps, mistake splurges, etc.) or no sound. I recommend sound-on as it gives you some indication of progress especially during a long save routine. The program adjusts itself for either 9 pin or 24 pin printers once you have given it your specifications. This ensures that you get an undistorted printout whichever you use. However there is a further utility which allows you to make additional modifications to your graphics files. This runs as a separate task to the main TnG program. Among other things it allows you to convert from 9 to 24 pin printing if you subsequently find that you need to do this.

It also gives options such as changing the print density (the number of passes the printhead makes) or changing the default centre justification of the graphics printout to left or right. This is particularly useful if you want text either side of the graphics. If you do want this you have to be clever with the control codes you choose in your _lis file as you have to judge the right number of reverse linefeeds and determine the new left and right margin settings.

A further benefit of TnG is that it will allow you to print out all those special and foreign characters that your existing printer driver won't pick up. Alan even gives an example of how you can print out true mathematical fractions by using special methods to control the printhead (superscript, backspace, normal, backspace, subscript etc.).

Not QPac2

The only real problem I had was when running it as a task within *Qpac 2* and this only concerned the option to 'grab' screens from other graphics programs multitasking with it. The result of a telephone conversation with Alan Bridewell was the conclusion that Qpac 2's Window Manager would not allow it.

All in all TnG is a very impressive package giving high quality printouts without making a major switch to a full blown DTP package. If you're a relatively inexperienced user you will need to spend some time becoming familiar with the various processes, as well as understanding the methods of operation. But I think you will get there, especially if you follow the manual. And if you do it is possible to get printouts which are quite stunning.

QMATHS 2.10

QMaths is in its second incarnation. Hilary Snaden adds up its functions.

INFORMATION
Program: QMATHS Maths System Part Two
Supplier: Digital Precision Ltd., 222 The Avenue, Chingford, London E4 9SE.
Tel. 081 527 5493.
Price: QMATHS part 2 £59.95. QMaths part 1 £69.95. Both parts £99.95.

Qmaths is the collective title as well as the "star component" of one of Digital Precision's most important recent software releases. The Qmaths program itself is described as a "symbolic algebraic expression simplifier". More simplistically, it's a sort of calculator which works with algebra.

As an alternative to the supplied boot program, Qmaths may be multitasked from Basic using EXEC, as it is written in SuperBasic and compiled with *QLiberator*. Unfortunately, while the QLiberator runtimes have been included, either they or the compiler used were not fully compatible with my Minerva rom as Qmaths could find neither Minerva screen when in two-screen mode.

In multitasking Qmaths from SuperBasic, the cursor kept switching itself back on after every operation from SuperBasic, but apart from that, and in Minerva's single-screen mode, it behaved well. Clear, on the main menu, redraws the whole screen as well as clearing the window.

The working

At the start, the QL screen is in three parts: a menu line at the top, a status line at the bottom, and the window displaying expressions entered, results produced and, as an option, the working.

The status line shows whether the printer is in use, what level of working has been selected, how long the last calculation took, and, when the program is working on an expression, what it is doing. R indicates data being read, . (full stop) an expression being scanned, T a transform in progress, and I that things are being shuffled to make more memory available.

User- and program-generated expressions are displayed in white, working in red. Each expression entered by the user or produced by the program is tagged with a sequential number so that previous expressions may be recalled, edited, and used again or as part of a further expression. The number of expressions kept available for recall depends on their size, but typically around four hundred can be entered before the earliest drop out.

The first menu option is Define. This brings up a sub-menu window (which restores the original screen when it disappears) into which one enters the expression to be worked on. To recall an earlier expression, type # plus its identification number.

Maths exams

As an illustration we will work through a problem at around GCSE/A-level standard, viz.

"Show that SIN(c)+SIN(d)=2.SIN((c+d)/2).COS((c-d)/2)"

An expression is entered on the "Define" menu something like:

```
2.sin((c+d)/2).cos((c-d)/2)
```

If graphics mode has been selected (which is how the program starts) this will be displayed in traditional algebraic notation. The working menu is selected by pressing W, and F (for full working-out) or T (to show transforms only) from the sub-menu which appears. It is also possible to select from sub-menus. If the printer option has also been selected, working will be sent to the printer as well.

The Printer menu offers a number of printer drivers. Special algebraic characters can be printed as dot graphics to Epson printers, as a custom character set for those capable of accepting a user-definable font, or as IBM graphics characters. Everything can also be printed in plain text; this will not be as neat as algebraic notation but it is a useful last resort if your printer will only accept Ascii characters. Basic source code for the drivers has thoughtfully been provided.

Simplify

The second menu option is "Simplify". This looks for parts of an expression which can be calculated or combined. Hence, if "Define" has been used to enter the expression

```
dif(cot(x))
```

where dif means the differential (with respect to x: this can be changed from the "Options" menu) of the expression in brackets, invoking "Simplify" will produce

```
-COSEC(x)^2
```

From the "Options" menu one can also free pi, e and i to have their meanings redefined; alter the base of logarithms; set memory size to "large" or "small"; change the multiplication symbol used, and select graphics or text mode. A "large" memory configuration is only essential when matrices as large as 4x4 are being used, but may be necessary if very large expressions are being manipulated.

Integration of expressions can also be attempted via "Simplify", but as anyone who has been involved with calculus will understand, there will be occasions when Qmaths will not be able to cope.

Solve

The Solve function is powerful; suppose we need to rearrange

```
x^2+y^2=(r^2/h^2).(h-z)^2
```

in terms of z. Using Define to enter

```
solve(x^2+y^2=(r^2/h^2).(h-z)^2,z)
```

followed by invoking Simplify produces the result

```
z=h-SQRT(((x^2+y^2).h^2)/r^2)
```

Qmaths is, however, unable to solve quadratics or solve an expression for a variable which appears more than once, such as

```
M=E-e.SIN(E)
```

which, as students of astronomy can confirm, cannot be solved for E except by an iterative method.

And the rest

Other functions available, besides trigonometric and hyperbolic trig, include logarithms, combinations, matrices, substitution and Taylor's expansion. In keeping with mathematical tradition, variables used by Qmaths, unlike those used by SuperBasic, are case-dependent.

To return to our original algebraic problem, the next step is to go to the Transform menu, which offers various ways of rearranging expressions. The obvious choice for this particular problem is "Trigonometric", since we are dealing with SIN and COS. Other options are "Algebraic", "Expand" and "Factor", each of which have sub-menus. From the Trigonometric sub-menu we choose "Sums & Multiples", and yet another sub-menu asks which direction the transform should take. For this particular

problem we would choose "Reverse".

Qmaths can now go to work. The time taken varies greatly depending on the size and complexity of the expression, the type of transform, and how much working is being shown. Using a printer, particularly in graphics mode, without a fairly generous buffer, will slow the program further. This transform took nine seconds with no working shown, and twenty-five with full working selected. Gold Card owners can expect dramatic reductions in these timings.

We are now presented with

```
2(SIN((c+d)/2+(c-
d)/2)/2+SIN((c+d)/2-(c-d)/2)/2)
```

an expression which means the same as the one we entered, but in a different arrangement. Expressions, especially expressions like this, look a great deal clearer when displayed in graphics format.

We now need another transform, this time going to the "Factor" sub-menu and selecting "Anything". Factorisation can be a slow process, and this one took eleven seconds with no working, 106 seconds with full working shown.

The result produced by Qmaths is

```
SIN(c)+SIN(d)
```

which, of course, is what we were trying to prove. QED.

Teacher!

The "working-out" is the program's, and not what a human brain is likely to produce, though it is a useful guide if interpreted carefully. Be aware that any maths teacher would spot the difference a mile off!

Expressions accumulated during a working session can be saved to disk for later use. There is no way of deleting unwanted expressions - mistakes, perhaps - and a useful addition would be a supplementary program which can edit and reformat the files, particularly for users who want to maintain a library of expressions.

The "Review" facility goes half-way there, stepping through previous expressions which can be reselected, or printed out either singly or in blocks. Strangely, there is no "Zap" command to clear out accumulated expressions: to restart with a clean sheet one has to Exit and reload.

There is no facility to export expressions or workings to a file. It would be useful to be able to incorporate this data in documents, but at present the only way is a roundabout (and undocumented) method of rewriting a printer driver with a filename in place of "ser1".

Export limits

It may be expecting too much to export graphics characters, since to print them from a wordprocessor would need either a custom font with text87 driving a printer in graphics mode, or a custom font and custom printer driver with Perfection. However, exporting the data as plain Ascii would be useful and it is difficult to see why this is not an option.

It is easy to become puzzled with a program of this complexity, and as well as a clear and helpful manual, context-sensitive help is available by pressing F1. The manual seems to have fallen a little behind the program in its assessment of memory requirements: with a code size of 147K and a dataspace of 246K Qmaths is (in QL terms) a very large program.

The manual quite reasonably points out that Qmaths by itself is unlikely to teach you algebra. If you are starting from scratch on a mathematical adventure you should get a good introductory textbook.

Having said that, Qmaths also has "lessons" on disk. The subjects begin with Basic algebra, and go through trigonometry and complex numbers to special relativity.

When a lesson is called up, text messages and simple diagrams appear on-screen explaining what's going on, together with embedded sample problems which Qmaths then proceeds to solve. The pull-down menus come and go so quickly that it is often difficult to see exactly how the program is tackling each problem, but it is usually possible to make a sensible guess at what is going on.

Lessons

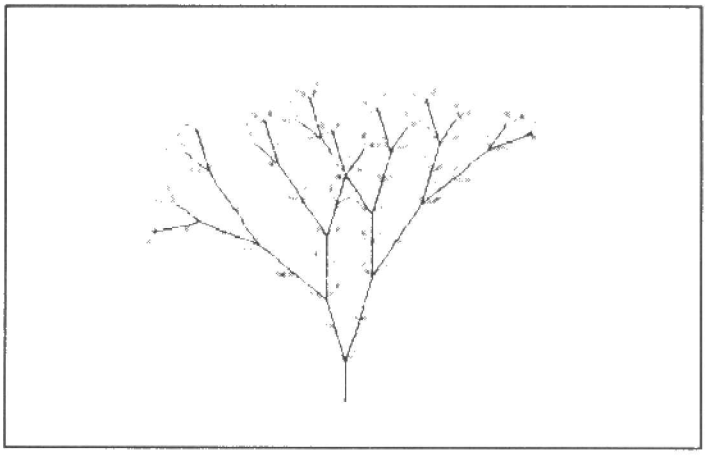
It is also possible to write your own lessons to save for future use. The lesson facility is excellent, enhancing the educational potential of the program. The lessons supplied may be a little concentrated for the newcomer to this type of maths; perhaps a wider range will become available.

Qmaths is not idiot-proof; that would be expecting far too much of any program which tries to do what Qmaths does. It is certainly likely, if used wisely, to help grow a "feel" for algebra, and could prove a valuable learning tool up to GCSE and A-level standard as well as for those wishing to explore this branch of mathematics. It may also be useful to people with an interest in engineering and astrophysics, where calculus and algebraic expressions needing rearrangement are intimidatingly frequent.

Qmaths is unique among QL software. The author deserves great credit for the thought and programming time which has gone into it.

QFRACT 1.06

Fractal script is a language used to define fractals so that the necessary complex maths are kept at a safe distance from the programmer. Qfract is a fractal script interpreter, taking programs written in fractal script to turn into graphic images.



Qfract is Turbo-compiled SuperBasic, requiring a short Basic extension of its own as well as the Turbo runtimes (all of which are supplied on the Qmaths disk) and at least 384K ram in which to run. The version of Qfract supplied for review would not run in Minerva's two-screen mode.

For some reason Qfract alters SuperBasic's priority to 1. This seems pointless as, since Qfract is run with EXEC_W, SuperBasic is suspended anyway. It be better if SuperBasic's priority was restored to its usual value of 32 on quitting Qfract.

One of the most striking features of Qfract is its custom character set, something one wishs for more often, since the QL's native font is not crystal clear. Menus appear at the bottom of the screen, leaving a large area for use by the program editor, and the whole screen is used for displaying the fractal images. The main menu gives these options:

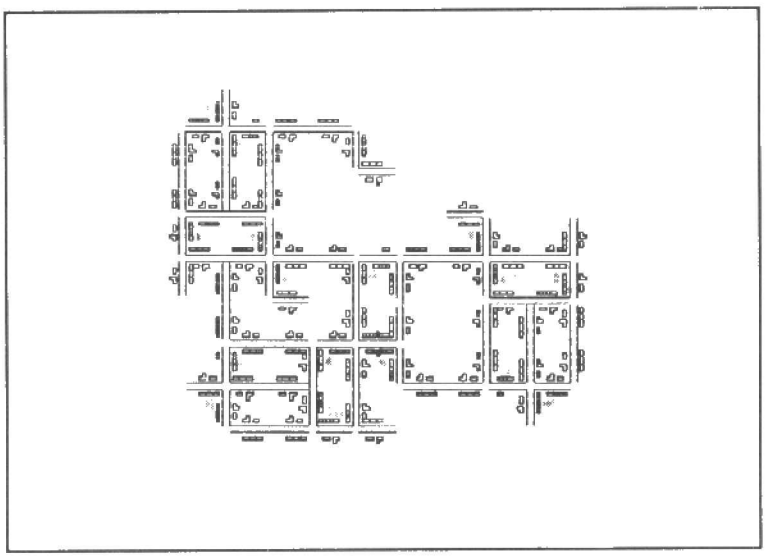
ESCape exits the program
F1 clears the contents of the program editor
F2 accesses the files menu
F3 enters the program editor
F4 redraws the image
F5 runs the program currently in the program editor.

The files menu allows one to save, load or delete program files, change the data device, and view a file without loading it. Program filenames end in _QFS, and only these are shown when a list of files is called up.

Unfortunately, when the Files - Load option is invoked there is a long wait while the program reads the disk and sorts out a directory, as a list which slowly scrolls under a static cursor; it looks neat but takes ages to select a file at the bottom of the list. QL users without a Gold Card may find the files menu a more taxing test of their patience than waiting for the fractal programs to run.

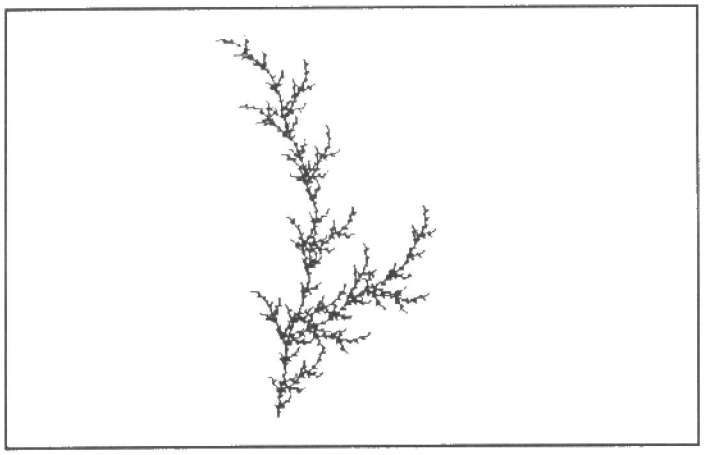
One cannot set different devices for reading programs from and saving screens, to use, say, flp1_ for programs and flp2

for screens one has constantly to alter the device via the device select option. One wonders if the files menu expects every user to have a Gold Card and hard disk or ED drives.



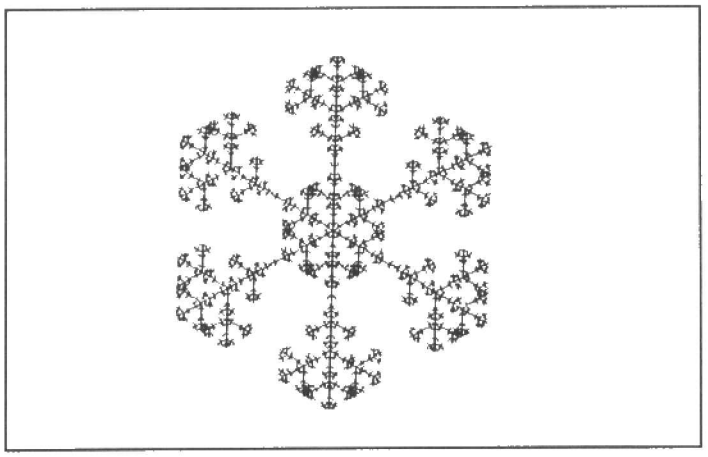
Delight

On the other hand, the program editor, an integral part of Qfract, and custom-designed for editing fractal script programs, is a delight. It is no more complex than is needed to edit the (usually) very short programs. Should users wish to prepare programs in their own, separate text editor this may also be possible.



Qfract accepts standard notation for the script, so programs additional to the generous selection supplied on the disk can be typed in from other sources with the minimum of modification. It is almost as easy to write one's own, following the copious information on fractal script (including a learn-as-you-go tutorial) in the manual, and easier still to modify the supplied programs. It is fascinating to see how much difference can be made by changing a single part of the original program.

The "Run" option uses the program in the editor to generate a "result string", the characters in which refer, essentially, to turtle graphics commands. This can take some time. A status window gives some idea of how far it has got.



Attempts to run the supplied example programs occasionally failed with a "multiple decimal points" error message which, on examination of the program code, seemed inexplicable. This was probably the most serious problem I had with any of the Qmaths programs.

Redraw

The length of result string is effectively limited only by the ram available, and the time you are prepared to wait. The result string is then converted into graphics commands which Qdos can understand and use to draw a picture. The "Redraw" option repeats this process, though as this can be tedious it is difficult to see why the current picture is not also stored as a complete screen image, which would allow the option of an instantaneous redraw.

A further puzzle, especially as one cannot be sure what a picture will be like until Qfract has finished drawing it, is that while it is possible to dump the screen to a printer, or to save the screen image as a disk file, these can only be done from within a script program. This is irritating if one has waited half an hour for a program to run, and forgotten to include the ScreenSave command.

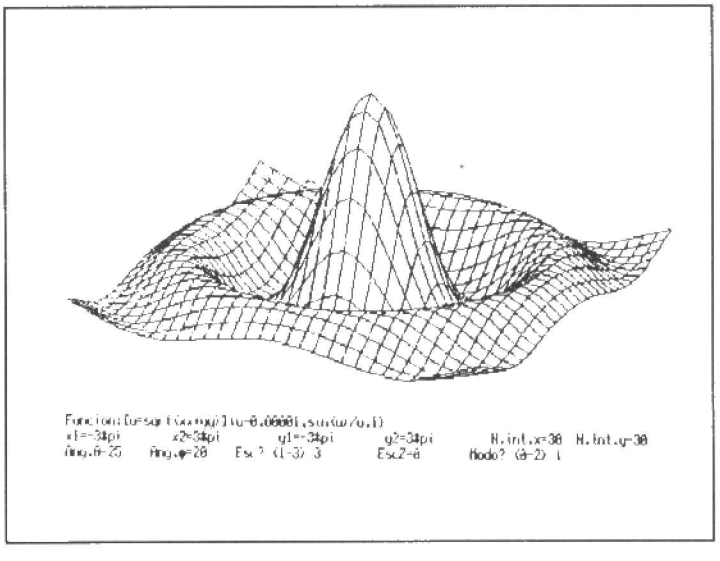
It would also be helpful, when the AutoScale facility is used within a fractal program, to be able to find out the scale value that Qfract chooses so that an appropriate scale can be written into later versions of the script program.

The manual deserves special praise as a clear and informative introduction to fractal script as well as a lucid explanation of the program.

QL users are likely to find fractal script addictive; the ability to produce lifelike images of trees and snowflakes from a few short lines of code deserves interest. Despite some irritating features, the quality of the graphics is limited only by the QL's current hardware, and the Qfract is an enterprising attempt to make a fascinating aspect of fractal mathematics accessible.

SURFACE MODELLER 3.0

Surface Modeller is essentially a program for drawing sophisticated 3D graphs. The program has a primitive but functional user interface - there are no menus as such at all - and the manual is necessary to master the program. The manual relates that the author wrote his own drawing code when he found QDOS's inbuilt line-drawing routines too



inaccurate. It has certainly worked, as and the program is capable of producing quite superb graphics.

There is no screen dump facility, but pressing P prompts for a filename under which the screen can be saved to disk, with or without a data caption, for subsequent printing by a utility such as Sidewinder. Pressing R redraws the screen in a wire-frame format.

Enough functions are available to cope with most calculations, and the results are stored on a stack, available for recall.

The data which needs to be entered is the function f(x,y) which is to be drawn, the ranges of x and y, the number of intervals in the x and y axes used to draw the graph, the angle of view, the scaling mode (none, automatic or manual), the scale if manual scaling has been selected, and presentation mode.

The manual says little about presentation mode beyond suggesting experimentation; there are nine modes (plus an undocumented mode 9 which appears only to clear the drawing screen) some of which seem to be fast draft modes while others are slower but more accurate. Coloured or wire-frame presentations are also available.

Speed is as impressive as graphics quality, even on a humble Trump-carded QL without the Lightning graphics extensions and in one of the slower presentation modes.

This is another program which will not run in Minerva's two-screen mode. In addition, the custom extensions, called QLVAL_EXTS on the disk, trash any default device definitions set up (via DATA_USE etc.) by Toolkit 2. However, as long as Toolkit 2 has been installed (with the command TK2_EXT on rom-based versions) the part of the boot program which says

```
CALL _addr+1444
```

can be removed. This does not affect the program but ensures that default devices are left intact. This is the sort of information which should be somewhere in the manual or the boot program but is not.

Surface Modeller has specialised applications, but it is also fun to experiment with; it usually recognises data input likely to produce a fatal error, such as divide by zero, and asks for a retry. Occasionally an error message appears, for reasons not always obvious, but usually traceable by close scrutiny of the input data.

Some users may prefer a neater (or more ostentatious) user interface, but what counts most - the graphics output - is outstanding.

HIGH PRECISION CALCULATORS 2.01

The most purely arithmetical program on the Qmaths disk is the high-precision calculator set, six in all, giving an accuracy of 14, 33, 72, 149, 303 and 611 figures, but otherwise identical.

The programs need input in Reverse Polish Notation; this means in essence no more than entering operators and operands in a different order, and one quickly becomes accustomed to it. *(Even I - Editor.)*

```
pi according to High Precision Calculator

(611-figure version)



3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344612847564823378678316527120190914564856692346034861045432664821339360726024914127372458700660631558817488152092096282925409171536436789259036001133053054882046652138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277053921717629317675238467481846766940513200056812 72
```

For example

```
1 acos <Enter>
```

calculates the value of pi.

Logically, the higher-precision calculators take longer; with a Minerva rom and Trump Card the above calculation takes seven minutes to 611 figures. Only the most specialised applications will need such accuracy, and the same calculation to a more reasonable 14 figures is almost instantaneous.

Enough functions are available to cope with most calculations, and the results are stored on a stack, available for recall. The programs multitask from SuperBasic easily using EXEC, and typing ? or HELP at the prompt brings up one of several inbuilt pages of handy hints.

All the calculators can read instructions from and save results to a file, so one can set up long-winded series of calculations and leave the program to get on with them.

Anyone with a need for maths to a higher accuracy than the QL's rom maths routines support could do little better than get a set of these programs.

As one might expect from Laurence Reeves, author of the Minerva rom, the calculators are fully compatible with Minerva.

Compendium

As well as the foregoing, the Qmaths disk contains several short Basic examples of how to deal with polynomials, statistics, matrices and various other problems. The manual invites users to use these in their own programs.

As a final comment, the Minerva rom has been around long enough that one wonders why new software is not fully compatible with it. Since this should involve little more than recompiling, Minerva users might reasonably expect updated, Minerva-friendly versions of the Qmaths programs to appear.

This is the kind of enterprising and imaginative software we could do with more of, and despite any reservations I have expressed about the presentation of the programs all of them do what they claim to, and do it well.

Speed is as impressive as graphics quality, even on a humble Trump-carded QL without the Lightning graphics extensions and in one of the slower presentation modes.

QDversion5

QD, the pointer-driven text editor from Jochen Merz, has now reached version 5. Wolfgang Lenerz investigates what is new.

INFORMATION
Program: QD5
Supplier: Jochen Merz, Im Stillen Winkel 12, D Duisburg 11, Germany
Price: DM 125

Before delving into QD, I should probably state that Jochen Merz, author of QD, is also distributor of a program I have written (FiFi, reviewed in the February issue of QL World). I have no other connection with Jochen Merz.

QD was, and is, the only text editor conceived to work under the pointer environment and take advantage of the its possibilities, so it can be used with a mouse, has movable/sizable windows, uses the (to my mind) easy-to-understand pointer interface menu system, is colour-configurable, and so on. As it has now reached version 5, a short review is in order. Indeed, if I follow the program author's version numbering system correctly, each new version corresponds to a major overhaul of the program, generally implying the addition of various new facilities. In this review I shall concentrate on the new(er) possibilities.

One major feature of QD is the ability to call upon a "Thing" directly from within QD. (Things are machine code routines which may be included in the QL. A list of Things is then maintained by Qdos, and Things can be used from within application programs - even Basic). The possibilities for using Things has existed since version 2 of QD, but it has now been extended to such an extent, that it is entirely possible for the Thing to take over QD completely. For example, the Thing can obtain lines from the file QD is currently editing, change them, and give them back to QD.

Things

The Thing to be used by QD must have a special format (the details of that format can be obtained free of charge from Jochen Merz). Once such a Thing exists it can be used from within QD, by hitting F10 (Shift-F5). The action it performs depends on the Thing itself. One such Thing is the QDasm Thing, an extension for users writing assembler code. This is on the QD disk, but users are not obliged to use it if they do not wish to.

This Qdasm Thing greatly eases the assembler programmer's burden: it is possible, for example, to format code automatically. Here, we see the incredible integration that is possible between the Thing and QD: when one uses the QDasm Thing in QD, QDasm opens a new menu within QD, one of the menu options being to format lines of codes (see **figure one**). Having selected this option, whenever one enters a line of assembler instructions, the line is reformatted according to the parameters set by the user. One can also call the QMAC macro-assembler (now sold by Quanta), or indeed other assemblers, directly with this Thing. This will save the file, call the assembler, have the file assembled and come back to QD with eventual errors - all in one go!

As can be seen from the above, QD is very much a programmer's text editor. To my mind, that makes sense, as I have never really been able to use text editors to write letters or other documents. Here the facilities offered are conceived with the programmer in mind (for example, there is no automatic line wrap, which is useless for writing programs anyway).

Search

The search and replace options have been quite extended compared to earlier versions: one can now determine up to four search/replace strings (but one can only use them alternatively, not concurrently). A nice feature is that hitting Ctrl-W when the cursor is over a word makes that into a QD "word" (actually a block), and one can then quickly set this to be the search/replace parameter. This is not to be under-estimated: how often does it happen that one wishes to see, for example, where a label is used, or an EQU substitution, or where a sub-routine is located in a large file? All it takes now is Ctrl-W on the label, call up the search menu, type 'T' (for 'Take word') and the search can starts.

It must be said, however, that for this type of operation, QD has a faster way: the GOTO menu where, indeed, it now is possible to go directly to an assembler label (see **figure two**): if one chooses 'assembler label' in this menu, QD displays a list of all the assembler labels in a window, and selecting one will position the cursor at that label. One gripe here: I would prefer the labels displayed to be ordered alphabetically, instead of having them displayed in the order in which they are in the file. I remain unconvinced by the author's argument that one generally knows whereabouts in a file the label will be (more towards the end, or at the beginning, and so on) and that, thus, one finds it faster in the list. Still, this feature is one of my favourites.

It is also possible, in the same fashion, to go to Basic functions and Basic procedures, if one edits a Superbasic program: selecting any of these options results in the list of procedures or functions being displayed, choosing one function then places the cursor at the line where that procedure/function is defined. So the Basic programmer has not been forgotten - and this is even more true with the all new "hypertext" help function.

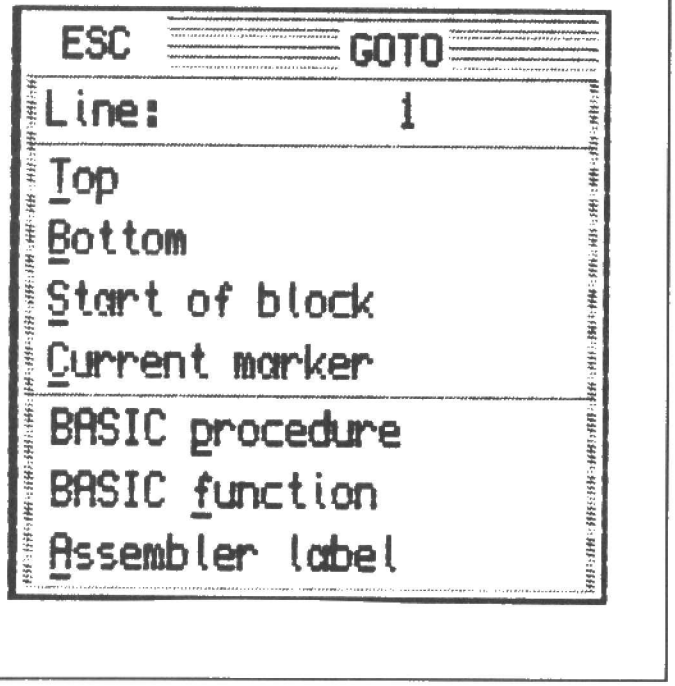
Hypertext help

Imagine: you are writing a Basic program in QD. There is a keyword you need, the syntax of which you can't recall (just where does the colour parameter go in the 'BLOCK' command?). Wouldn't it be nice to put the cursor over the word 'block', hit F1 (the normal help key) and have a page of text

come up, telling you all about this keyword? And so it does (see **figure three**). QD now has an extensive and expandable help system, which can give help about anything, provided the help files and an index file for it exist and are in a certain format. The help files for the Basic keywords are of course supplied with QD, so you don't have to type them in yourself. For assembler programmers, a Basic program is supplied that can make a help index file of all XDEFs in assembler files - that way, doing "help" on a label of an external routine will call up a help window with that routine!

Some other new aspects of QD 5 concern the menu extensions. The menu extensions are, of course, a separate piece of software, even though QD will not function without them. Their purpose is to supply a certain set of standard (file) functions to all programs "aware" of these extensions. Technically speaking, the menu extensions are a "Thing" (but not a QD Thing) and are loaded in the boot via a RESPR and CALL combination.

The menu extensions were extensively reviewed in the August 1992 issue of *QL World*, so I shall only talk about them here insofar as they are visible to the QD user and sport new facilities. For example, whenever one has to select a file (to load or insert), another window is opened where one can comfortably choose the file to be loaded. This "file selector" is actually one of the menu extensions. A new - and very welcome - option in the file selector box is the possibility to "view" files before loading. If one chooses ("hits" in pointèr-environment-speak) a file when the "view" option is selected, then one can see this file before loading. If one confirms that this is the file to be loaded (a "do" on the file) then the file is loaded. This is very straightforward and easy.

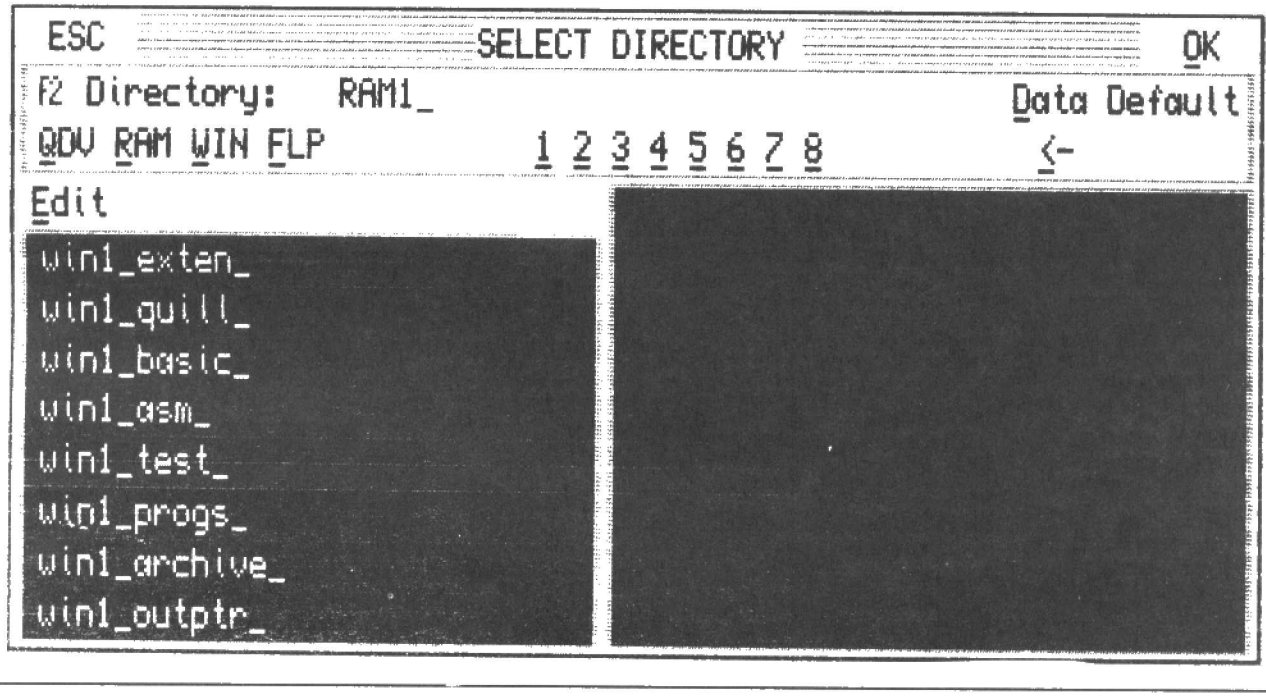


Directories

Another new option can be found in the "directory selector", from which one can choose a directory for file operations. This directory selector allows a quick choice of directories, and proposes a certain number of predefined sub-directories. Of course, the user can configure these pre-defined directories before loading the menu extensions. Up now it was not possible, however, to re-configure the pre-defined directories once the menu extensions were loaded. A new "edit" option does now allow this, and it will then save the newly configured menu extensions into a file of your choice.

The menu extensions also contain another set of extensions, the 'scrap' extensions. The idea of the scrap extensions is that, like the Macintosh scrapbook, it should be possible to put data onto the scrap from within one application, and then put it back into another application. Of course, QD now supports this and it is possible to put all or part of the file onto the scrap and/or get it from the scrap. This is actually the fastest way to get some part of a file from one QD to another: put it on the scrap from within one QD, and then get it from the scrap into the other QD. This is just like a 'cut and paste' operation on other (lesser!) computers. It is to be noted that other software (such as Qspread) is starting to use the scrap, so that it is becoming possible to cut and paste between different applications!

I have only described the major changes to QD. I have not described things like the fact that it now can have shorter button names. You will probably have guessed from the general tone of this article that I am enthusiastic about. As I tend to do a lot of programming, I make extensive use of QD, and it is a program I can recommend wholeheartedly.



SuperBasic in Action

Simon Goodwin's PAGE PRINT gets user-friendly.

Last month I introduced PAGE_PRINT, a SuperBasic program to pack pages of text onto one sheet of paper. The first version produced impressive results, but had a very simple user INPUT routine, which is fine if you know what you're doing but not to everyone's taste.

This month's listing extends PAGE_PRINT with a friendly menu which shows the size and format of each potential layout, accepts any page length, and prints helpful messages as it goes along.

The program illustrates ways to implement a menu bar, validate entries and generate italic headings in SuperBasic. It uses windows #0 and #1 and a compiled version will happily multi-task without assistance on any QL thanks to the useful but undocumented PAN #0,0,115 instruction at line 2290, which turns on a cursor before INKEY$ calls.

I, J and K are only used to hold whole numbers, typically array subscripts in FOR loops, so they are best defined as integers if you wish to compile the program with Qliberator or Turbo. The resultant code will be faster and shorter as a consequence.

Splicing

The accompanying listing will not run on its own, but only in conjunction with last month's Basic. The new lines fit before the others, linked by the revised PROCedure USER_INTERFACE, which now calls CHOICE% rather than present a series of INPUT questions. Change the lines to match the second listing if you want to run both parts together. The old INPUT lines are still there, but have been 'commented out' with REMark keywords. An added IF on line 2930 prevents the Epson initialisation string appearing in condensed or plain Ascii output, as this may confuse some vintage printers.

CHOICE% starts by drawing a shadowed window with a border suggesting light from the top left corner. The heading should stand out without appearing to shout, so I use a sequence of WINDOW and PAN commands at line 1230 to slant the text evenly to the left, creating a new print style without changing the fount.

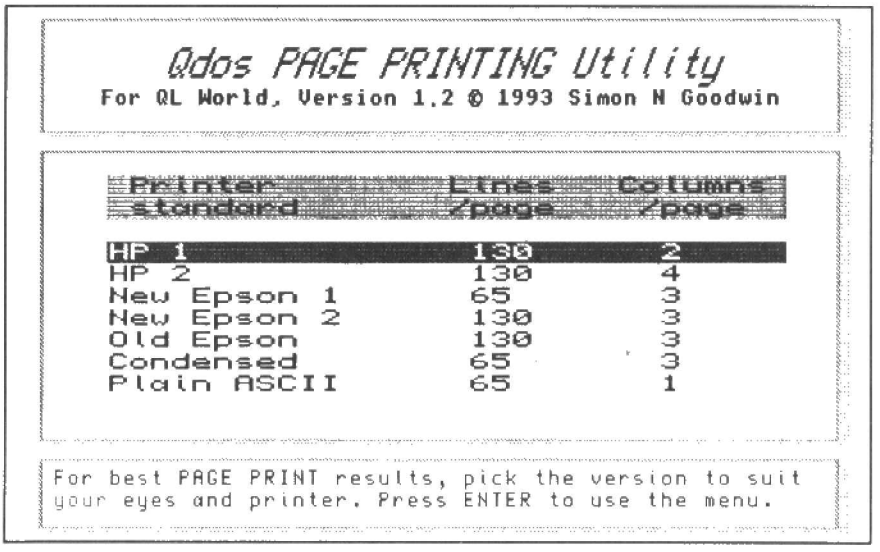
The hard-working default window is then moved down the screen, where a second shadowed box will contain the main menu. Note how channel #1 can be moved around to get the appearance of several windows, without wasting memory by opening them all at once or confusing the program with a plethora of # symbols.

The greatest improvement over the INPUT version is the menu which shows the seven print styles, indicating the number of columns available in your chosen width as well as the printer details. The PAUSE before each KEYROW call prevents the menu bar from moving too quickly on a Gold Card or other accelerated system.

The filename is not checked, but Toolkit users could add a call to FSTAT or DEVICE_STATUS, with

```
        QL World SuperBasic in Action, PAGE_PRINT part 2, May 1993

1000 REMark PAGE_PRINT 1.5 by Simon N Goodwin, 6th May 1993
1010 REMark MERGE with the program on pages 32-34, April 1993
1030 :
1160 DEFine FuNction CHOICES%
1170 min_height%=10 : max_height%=4000
1180 MODE 4 : WINDOW 452,52,36,18 : PAPER 2,0 : CLS
1190 WINDOW 452,52,30,15 : BORDER 1,4,2 : PAPER 0 : CLS
1200 CSIZE 0,0 : AT 1,11 : CSIZE 2,1 : INK 7
1210 PRINT "Qdos PAGE PRINTING Utility"
1220 REMark Make italic banner
1230 FOR i=1 TO 9 : WINDOW 350,2,80,25+i*2 : PAN 10-i
1240 WINDOW 448,50,32,16 : CSIZE 1,0 : AT 3,4
1250 PRINT "For QL World, Version 1.5 © 1993 Simon N Goodwin"
1260 REMark Add shadow to main window
1270 WINDOW 452,132,36,78 : PAPER 2,0 : CLS
1280 WINDOW 452,132,30,75 : BORDER 1,4,2 : PAPER 0 : CLS
1290 INPUT \" Input file (e.g. FLP1_BOOT): ";file$
1300 IF file$="" : PRINT #0;"Cancelled." : STOP
1310 REMark Add FTEST or DEVICE_STATUS here if required
1320 REPeat get_width
1330   PRINT \" Column width (e.g. 40): ";
1340   cols_used%=POSINT(#1)
1350   IF cols_used%<1 OR cols_used%>127
1360     PRINT \"Please specify a width between 1 and 127."
1370   ELSE
1380     EXIT get_width
1390   END IF
1400 END REPeat get_width
1410 col_spacing%=cols_used% + column_gap%
1420 WINDOW #0;452,32,36,218 : PAPER #0,2,0 : CLS #0
1430 WINDOW #0;452,32,30,215 : BORDER #0,1,4,2 : PAPER #0,0
1440 CLS #0 : BORDER #0,4,128 : INK #0,7 : CSIZE #0,1,0
1450 :
1460 CLS : PAPER 2 : INK 7 : CSIZE 2,0
1470 AT 1,3:PRINT " Printer        Lines   Columns "
1480 AT 2,3:PRINT " standard       /page    /page  "
1490 i=1 : PAPER 4 : INK 0 : SHOW_VER$
1500 PAPER 0 : INK 7 : FOR i=2 TO pages% : SHOW_VER$
1510 PRINT #0;"For best PAGE PRINT results, pick the ";
1520 PRINT #0;"version to suit ";
1530 INK #0,4 : PRINT #0;"your "; : INK #0,7
1540 PRINT #0;"eyes and printer. Press ENTER to use the menu. ";
1550 SURE_PAUSE : i=1
1560 REPeat choose
1570   CLS #0
1580   PRINT #0;" Use the vertical arrows ( ↑↓ ) to choose a version."
1590   PRINT #0;" Press SPACE to confirm, or ESC to return to BASIC. "
1600   REPeat poll
1610     PAUSE : k=KEYROW(1)
1620     IF k=0 : NEXT poll
1630     IF (k && 128) AND i<pages%
1640       SHOW_VER$ : i=i+1 : PAPER 4 : INK 0
1650       SHOW_VER$ : PAPER 0 : INK 7
1660     END IF
1670     IF (k && 4) AND i>1
1680       SHOW_VER$ : i=i-1 : PAPER 4 : INK 0
1690       SHOW_VER$ : PAPER 0 : INK 7
1700     END IF
1710     IF k && 64
1720       CLS #0
1730       PRINT #0;"PAGE PRINT version " & i;
1740       PRINT #0;" selected - are you sure?";
1750       IF YEA OR NAY : EXIT choose : ELSE NEXT choose
1760     END IF
1770     IF k && 8
1780       CLS #0 : PRINT #0;" Cancelled." : STOP
1790     END IF
1800   END REPeat poll
1810 END REPeat choose
1820 CLS #0
```



```
1830 PRINT #0;"The default paper size is A4, with ";
1840 PRINT #0;page_length%;" lines per page."
1850 IF i>=epson%
1860   PRINT #0;"Do you want to change this";
1870   IF YEA_OR_NAY
1880     REPeat get_length
1890       CLS #0 : PRINT #0;'Enter page length, in standard ';
1900       PRINT #0;'3.1mm or 1/6" lines ?'
1910       page_length%=POSINT(#0)
1920       IF page_length%<min_height% OR page_length%>max_height%
1930         PRINT #0;"Sorry, this software expects ";
1940         PRINT #0;min_height%;" to ";max_height%;" lines/page."
1950         PRINT #0;"Press ENTER to try again or ESC to quit. ";
1960         SURE_PAUSE
1970       ELSE
1980         EXIT get_length
1990       END IF
2000     END REPeat get_length
2010   END IF
2020   REMark Work out gap between pages
2030   blanks%=5 * ( 1 + (i=4 OR i=5) )
2040   temp%=(height%(i)+blanks%)/70 * page_length% - blanks%
2050   height%(i)=temp%
2060 ELSE
2070   blanks%=0:REMark No explicit blank lines in HP mode
2080 END IF
2090 RETurn i
2100 END DEFine CHOICES%
2110 :
2120 DEFine PROCedure SHOW_VER$
2130 AT #3,3 : PRINT #3;ver$;
2140 PRINT TO 20;height%(i); TO 29;
2150 PRINT (width%(i) + column_gap%) DIV col_spacing% TO 35
2160 END DEFine SHOW_VER$
2170 :
2180 DEFine FuNction YEA_OR_NAY
2190 LOCal k$(2)
2200 PRINT #0;" (Y/N) ";
2210 REPeat loop
2220   k$=INKEY$(#0,-1)
2230   IF k$=="y" OR k$=="n" : EXIT loop
2240 END REPeat loop
2250 PRINT #0;k$ : RETurn k$<>"n"
2260 END DEFine YEA_OR_NAY
2270 :
2280 DEFine PROCedure SURE_PAUSE
2290 PAN #0,0,115 :REMark Turn cursor on
2300 PAUSE 5 : PAUSE 5 : PAUSE 5 : REMark Lose key bounce
2310 IF INKEY$(#0,-1)=esc$ : PRINT #0\\"Escape pressed." : STOP
2320 END DEFine SURE_PAUSE
2330 :
2340 DEFine FuNction POSINT(channel%)
2350 LOCal k$(6)
2360 INPUT #channel%;k$
2370 IF "e" INSTR k$ OR k$="" THEN k$="0"
2380 IF k$<0 OR k$>32767 THEN k$=""
2390 RETurn "0" & k$
2400 END DEFine POSINT
2410 :
2820 DEFine PROCedure USER_INTERFACE
2830 LOCal t%
2840 form% = CHOICES%
2850 REMark INPUT "File name     : ";file$
2860 REMark INPUT "Column width  : ";cols used%
2870 REMark INPUT "Format (1..7) : ";form%
2880 REMark blanks%=5 * ( 1 + (form%=4 OR form%=5) )
2890 REMark col_spacing%=cols_used% + column_gap%
2900 IF form%<epson%
2910   pr_init$=hp_init$
2920 ELSE
2930   IF form%<6 THEN pr_init$=epson_init$
2940 END IF
2950 page_size$=page_spec$(form%)
2960 t%=width%(form%) : REMark "JM" bodge
2970 DIM grid$(height%(form%),t%)
2980 BLANK_PAGE
2990 END DEFine USER_INTERFACE
```

parameter 1 for input, to check that the named file exists before continuing.

The number of lines per page is adjustable from 10 to 4000, which would gobble almost 800K per 'page' in the GRID$ array. The prompts refer to 'standard' line pitch of 3.1mm or one-sixth of an inch per line, but you can just type the number of lines available in your default fount and paper size. Unless you are using HP codes remember to allow for the five full-height blank lines at the bottom of each fan-fold page.

New Old Bug

PAGE_PRINT will run on any QL or emulator - it has been tested on JM, JS and Minerva Qdos, and uses only inbuilt features of Sinclair SuperBasic. In the course of testing PAGE_PRINT I discovered yet another bug in early Sinclair roms - you can see the work-around at line 2960 in the USER_INTERFACE procedure.

If you use an array value in the expression setting the length of a string array element, in DIM, the JM QL ignores the true number of elements and limits the first subscript to zero. Thus this program gives an 'out of range' error when it tries to print text$(1):

```
1 DIM limit(0) : limit(0)=802
DIM text$(9,limit(0))3 FOR i=0 TO 9 : PRINT i,text$(i)
```

The cure used in PROCedure USER_INTERFACE is to store the limit in a temporary variable and use that in the DIM, like so:

```
2 t%=limit(0) : DIM text$(9,t%)
```

This is not a serious bug once you know how to get around it, but it's rather confusing when first encountered. It also affects multi-dimensional arrays if any dimension after the first in the DIM is specified by an array.

More WorkPAGE_PRINT would be even nicer if it let you select a file name, or perhaps a set of names, from a menu. This brings SuperBasic In Action to the topic of the 'file requester', which will have to wait for another column. Till then, enjoy PAGE_PRINT and please write to me if you think of any other challenges for this practical SuperBasic column.

JOCHEN MERZ SOFTWARE

Software - Development - QL-Soft-& Hardware Distributor
Im stillen Winkel 12 - D-4100 Duisburg 11 Germany
Telephone & Fax: 0203 501274 - Mailbox 0203-591706

New Games: The Oracle (by Jochen Merz) The Oracle is an ancient tactic-puzzle, where you do not have to be fast but clever! You have to fill different tiles into a field, but there are various rules how to do it. If you can place all the tiles you a bonus - if you obeye special rules you get much more points. Every game is different, you'll never be bored. Features: high-score table, hints for the next move etc. **DM 49,90**

MineField (by Bernhard Scheffold) Another game for the Pointer Environment. You need skill and concentration to clear a minefield. Many options, configurable size, number of mines etc. Toolkit II required. **DM 39,90**

QD Version 5 - The first (and only) Editor using the PE. Dynamic memory allocation, no limit on numbers of lines, comfortable block-handling and many, many new features, e.g. improved print menu, better search/replace, GOTO Procedure and Function, even Machine code label.
New features: V5 Thing Interface which makes QD5 extendable and inbuilt **HyperText HELP System!** This HELP System is very easy to use, simply move the cursor over a word and press HELP (F1) and you get extensive help on the subject. With complete HELP (German and English) for SuperBASIC, inclusive examples! It is very easy to make Assembler-Help, e.g. get external definition of routines, libraries etc. with one keystrokes. HELP can be called recursively, which means, you can get help on previous help on another help subject. You can even edit the help texts, add remarks, examples etc. **DM 125,-**
Upgrade from QD V4 DM 30,-
All our products which contain SuperBASIC extensions will be updated so that they will have files which you add to your HELP System so that help is provided for additional Procedures/Functions!

HyperHELP BASIC
This product gives you instant help in SuperBASIC! The price is much lower for those who program in SuperBASIC only and do not require a full QD (although it is very useful). HyperHELP can be executed, put onto HOTKEY or woken from a Button. It displays the full set of SuperBASIC procedurs, functions and keywords currently existing in your machine, plus additional help on operators, identifiers etc. Simply click on the word you get the full description, use of all parameters plus examples. FORGET ABOUT YOUR MANUALS for parameter description! You get complete help on SuperBASIC, in German and English. The help files can be updated with any editor, Quill, Text87 so that you can update and add remarks whenever you want. **DM 49,-**

QMenu-the Menu Extension NOW IN VERSION 4!
QMenu is a very easy to use interface with pre-defined menus (e.g. multi-column file-select, simple-choice boxes, select from lists, error handling). These menus may be used from SuperBASIC, machine code and other languages. New features: the directory name is (optionally) not fully listed, i.e. only the names INSIDE the subdirectory are given, allowing for much more files to be listed in the window. New DO AND REPORT menu. Brandnew feature: pre-defined directories and Extensions may be changed and configured at run-time! More examples, improvements here and there, which makes the menu Extension getting mature. **DM 39,90**
Update with new manual DM 16,-

QSpread - a new Spreadsheet program. It is completely mouse- and/or keyboard-controllable and uses, of course, the Menu Extension. Windows may be enlarged up to the maximum screen area, it may be split in up to three different horizontal and vertical sections, giving 9 independent controllable areas. Every cell may be formatted independently, with many options (justification, decimal point etc.) and with preview. Sum- and other often-used macro-functions. Maximum: 32768 cells, where, for example, 16000 cells need about 400kBytes. Block handling and block entry is very easy, especially if you have a numerical pad: you select the block and enter the values one after the other. They are automatically place in the right order. No cursor-key-action necessary! New cellname-enquiry, echo-function, different rounding methods. Many addtional functions, which belong to today's standard-equipment: Help, Button-function, use of the Scrap, all standard file operations, calculation order row or columns etc. [V1.12] **QSpread with comprehensive manual DM 169,-**

QDOS Reference Manual DM 89,90		QPTR DM 92,-
SER Mouse driver DM 40,-		QSUP DM 79,90
FiFi DM 49,90		DISA DM 90,
QShang DM 45,90	Thing & EPROM Manager	DM 61,50
LineDesign DM 249,-	text87plus4	DM 239
Diamonds DM 35,90	BrainSmasher	DM 45,90
Arcanoid DM 35,90	Firebirds	DM 35,90
SuperGamesPack DM 90,-	The lonely Joker	DM 52,90
EasyPTR Part 1 DM 89,-	Part 2 DM 49,-	Part 3 DM 49,-

QVME - Emulator card for ATARI Mega STE DM 695,-
Harddisk-Kit for ATARI Mega STE DM 99,-
Medium for 44MB SyQuest changeable harddisk DM 143,-
All sorts of Quantum SCSI-harddisks at cheap prices available, spare ones or readily cased (with cables, adaptor etc) for ATARIs.

Please add DM 13,- for postage and package (Europe) or DM 13.- for one item and DM 7,- for every further item (Overseas). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE.

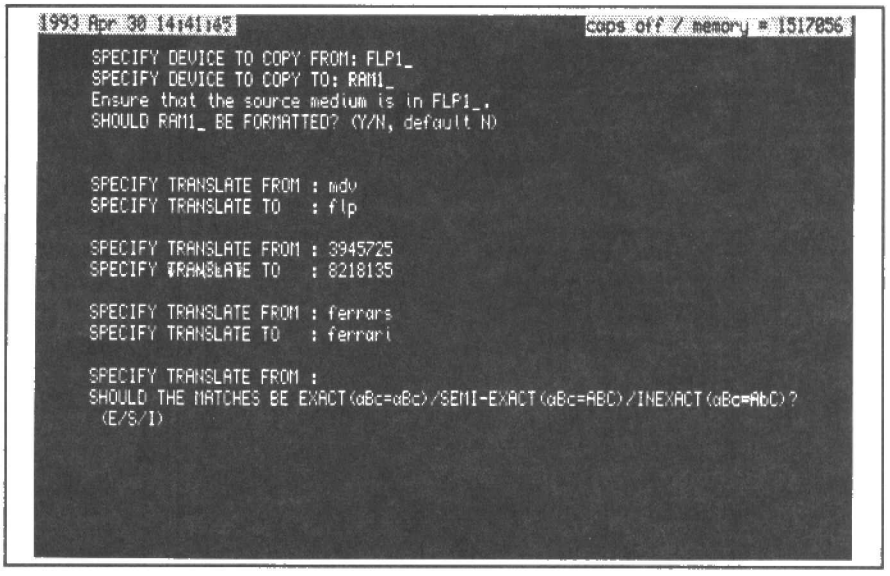
Eurocard Access MasterCard VISA Cards

Transfer Utility Special Edition

Bryan Davies tries the latest file-transfer update.

INFORMATION
Program: Transfer Utility Special Edition
Price: £29.95
Supplier: Digital Precision, 222 The Avenue, Chingford, London E4 9SE.
Tel. (081)-527-5493

The basic version of Transfer Utility was reviewed a little over two years ago, and is still available at the original price of £9.95. In its SE form, it is still a program without any superfluous screen dressing, but it has some useful (and interesting) new features.



The program is supplied on disk only. The instructions suggest you use the program itself to make a back-up copy of the disk. Being EXEC-able, it can be run alongside other programs. No restriction is placed on memory size, indicating that the program can be used on a basic QL. It works with any of the standard QL devices, so you can copy your old microdrive cartridges onto floppy disk, for instance. This is a good example of what the program might be bought for - transferring files from cartridge to disk and, at the same time, changing any drive references within the files from mdv to flp. The options available make the program useful as an organisational tool; you can re-arrange files on media and modify file names.

Upmarket COPY

Written instructions are, by Digital Precision's standards, very brief, running to just six and a half sides. They are quite sufficient. You can regard the program as an enhanced form of the COPY command, and reams of description are not required for it. While COPY has served us all moderately well over the years, the introduction of WCOPY gave us a much more useful version of the command. There have been a variety of other copy commands from other suppliers, but it is likely that WCOPY will be the only one known to most users. However, that command has some oddities. Transfer Utility does a better job than WCOPY, although it is not a direct replacement for it in everyday use, in so far as WCOPY is a resident procedure whereas Transfer is an EXEC-able program. If you care to experiment with the copy commands you have available, you may be surprised at the results; depending upon how unusual the file names are, you may find the number of files copied differs quite considerably from command to command. Transfer Utility is likely to "find" the most files. There is no intention of offering the program as a program duplicator; it will not duplicate the copy-protection "keys" used with some programs.

The SuperBasic extensions files required by Transfer are loaded via a boot file, and they are familiar ones - XTRAS, XTRAS2 and XTRAS3. These file names appear in other DP programs, and it is quite likely that your existing system boot loads the extensions Transfer requires, making this additional boot unnecessary. XTRAS is the standard set of Turbo run-time extensions, as used with The Editor etc., XTRAS3 is part of TK2 (Toolkit II), and XTRAS2 is a device for checking if TK2 is already present in the system and switching it on if it is. Thus, if you already use XTRAS, as is likely, and you have an interface such as the Trump or Gold Card and routinely use the command TK2_EXT (if it has to be used explicitly) in your boot(s), there is no need to load the extensions files supplied with Transfer.

What's it for?

The functions provided by Transfer Utility SE are straight device-to-device file or disk copy; string-translation; file name case changing; file sorting

The standard copy procedure includes the FORMAT command, in case your destination medium is unformatted, or simply needs emptying of files. Device names can be of up to 32 characters each, which should be sufficient for most hard disk users. The first illustration shows the screen after the first few steps have been taken; the devices to be used have been named, the offer to format the destination device has been turned down, three translate pairs have been specified, and the decision on the form of string matching to be used is about to be taken.

You are automatically asked, every time you make copies, if any character string needs to be translated to something else. If you specify one translation requirement, you are then asked if you want others, until you indicate that is all by pressing Enter. At this stage, there is no mention of case-dependence, but this question comes next. You are asked to specify whether you want the string matching to be Exact, Semi-Exact, or Inexact. Exact is fairly obvious - Transfer will ignore MDV or MdV if you specify mdv as the "Translate from" string. Semi-Exact is more subtle and is best explained by example. Specifying mdV as the "from" string, and Semi-Exact as the mode, causes mdV, MDV, mdv and Mdv to be found. That is, the specified string, and the string all in capitals or all in lower-case or with only the first character in upper case, is found. This is the recommended operating mode. Inexact is what you would expect - all combinations of upper- and lower case will be found. This mode needs to be treated with caution, as you can get unexpected (possibly undesirable) results from it.

The string search is not "word-

specific" - the specified string will be found both on its own (a "word") or when contained within a larger group of characters. This means that "mdv" will be found in both "mdv$" and "mdv1". As you would normally not want to do a global string replacement in such circumstances, it pays to do a bit of thinking before typing-in strings. The same applies when more than one translation is required, if there are common strings in two or more of them. For example, specifying the translation ABC to BAB first, and ABD to PQR second, would result in xxxABCDxxx ending up as xxxBPQRxxx. The program searches through the file from the start, making translations sequentially, then repeats the process for any further translate instructions. The "Translate to" string will always be inserted exactly as typed-in.

Name changing

Would you want to change the case within a file name? At first sight, you might think not, but there are circumstances in which it would be helpful to do so. You might prefer to see certain features of file names emphasised in a listing, such as having the extensions in capitals and the prefixes in lower case (quicklaser_CONFIGURE, chicago_F87). Sorting on file names will be affected by case, and it might make a subsequent sort more effective if file names were changed beforehand to all-lower or all-upper case. DP gives the example of files named "DOG", "Cat" and "bird", which would be placed in the order Cat-DOG-bird if a standard ascending sort on Ascii characters were carried out (character codes are in the order z > a > Z > A in the Ascii set). In case making changes to file names worries you, note that there is no apparent reason why the case used in any file name should affect its use in any way.

It may be hard to believe, but there are 16 possible ways of setting file name case with Transfer. The prefix and suffix (extension) parts of file names can be set - separately - to be in all-upper (AAA), all-lower (aaa), or mixed case (Aaa). Because of the Qdos oddity of allowing multiple usage of the underscore within a single file name, there can be problems when performing operations on files, but note that only the group of characters after the last underscore is the extension - "BAS" is the extension in Turbo_Config_BAS, and "_Config" is part of the prefix. Any odd files that you do not want treated in the same way as the others can be excluded during the course of the case-changing routine.

Sorting

The options offered are (by) Name, Extension, Length, name Length, Update date, Type, dataSpace, and Not sorted. When one sort has been completed, you are offered the chance to have another done, to further refine the process. Each sort can be in either ascending or descending order.

As noted above, it is desirable to consider changing file name case, to make the format of all names consistent, before doing a sort on Name. This applies with both prefix and suffix. Length refers to the number of bytes occupied by a file, name Length refers to the number of characters in the file name itself. The Update date is when the file was last altered, Type refers to the two standard Qdos file types (1 for executable files, 0 for all others), and dataSpace is the memory area reserved by some executable files for working on data (the value is assumed zero for all non-executable files).

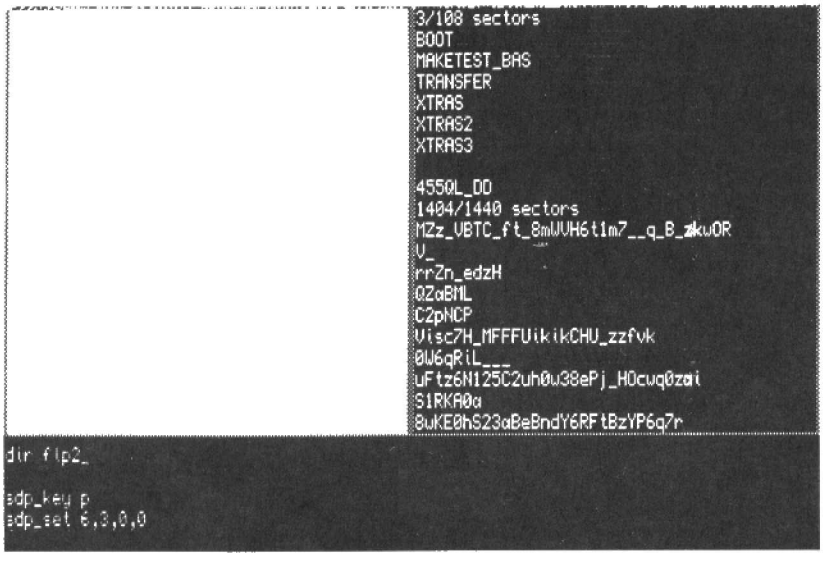
Multiple sorts are of considerable interest, and the instructions devote over a page to example sorts. As is pointed out, a second sort does not necessarily upset the order obtained from the first sort, and it is possible to sort according to different criteria on successive passes and obtain a multiply-ordered set of files. You could, for instance, change the case of file prefixes to upper, do an ascending alphabetic sort on file name, and a descending sort on file length, to get files of similar names grouped in inverse order by length. In doing such operations, you need to give thought to which order to do the sorts in, however. The examples show how carrying out sorts in the wrong sequence can produce results quite different from what is desired. There is a SuperBasic file on the supplied disk which will create a test disk to be used for experimenting with Transfer. As can be seen from the second screen shot, the file names on this disk are a trifle unusual! The upper part of the illustration shows the files on the supplied program disk, the contents of the test disk being in the lower part.

Sorting by file size is interesting, and can be useful too. A file on a disk that was copied during the review turned out to be of zero bytes length, which answered a long-standing question about why text87 Plus4 had been in the habit of saying one of its fount files was corrupt, during start-up. It is not too difficult to lose track of what is on a much-used disk, especially if it is an extra-density one, as in this case. Transfer Utility might be used here to create a replacement disk, on which the files are all in a specified order, to make it much easier to spot particular files in a directory listing.

Options

Throughout the on-screen prompts, default values can be accepted by pressing the Enter key, and the usual Y/N/A/Q (yes, no, all, quit) options are offered to allow selection. Once all your requirements have been specified, a final press of the Enter key starts the action. Each operation is tackled separately - once translates have been done, and the case of file names has been set, the first sort is performed, then the second and so on, with the actual copying being the final stage. You have one more choice, or set of choices, to make before copying starts. The first file to be processed brings up the message "Transfer file_name? (Y/C/N/A/Q)" - slightly different from the usual form, in having a C. This stands for Change and allows you to manually change the file name, of this particular file. Pressing Y, N or C permits you to make individual decisions on files as their turns come for processing; Q aborts the transfer process, A sets it to go through all the files without further interference from you. The overall process takes little time, even with a well-filled ED disk as the source (the review system had a Gold Card, though).

Transfer Utility Special Edition works well, and provides an interesting view of your files. It is without frills, which is as it should be for a "utility". No significant problems occurred with it during the review period; one stoppage occurred when the destination device became full, and the usual Turbo message "Task halted...drive full" appeared (Transfer apparently does not look to see if there is sufficient space available for transfers). One of the features that makes this program more than just a copying tool is the commentary that is presented on the screen as the transfer process is carried out. It is educational for the average user. As each file is copied, its (new) name is written to the screen, together with the file length, dataspace (if any), and number of translations (if any) carried out within it. One tends to ignore what is accumulating on disks - Transfer reminds you what you have got, and what it all is.



JDIR

Bryan Davies pages his DIR with a new utility.

INFORMATION

Program: JDIR
Price: £10 (no credit card sales)
Supplier: J J Haftke, 7 Lansdown Road, Sidcup, Kent DA14 4EF. Tel. **081 302 6154.**

There are now so many extensions to the basic QL set of commands that an outsider could feel the original operating system must have been sadly deficient. When you look through a list of these extra items, though, you might be hard put to identify many that you actually use. Quite a number of them are there for the convenience of application programs and will not be used directly from the command line. The extra commands you use regularly yourself may well be few - WCOPY, PROG_USE, EX, EW, etc.

acceptable, but the command structure is not set up that way. JDIR is an alternative directory-listing command, and a very handy one. Any enthusiastic computer user comes across dozens of programs, and discards many of them after a cursory investigation, but JDIR is one of the few which you might want to keep around. Too many of the programs we see are not good enough for long-term use, but cost £20 or more, which is serious money to waste for anyone living on a small pension. At £10, JDIR is in the

CALL sequence, from a boot file. The necessary SuperBasic lines can be incorporated into an existing boot file. Because of the shortage of microdrive cartridges these days, anyone wanting JDIR on cartridge will have to request it as a special order (and may have to supply the cartridge, presumably). One drawback is that the extensions loaded to make the new directory command usable take up about 18 KB of ram. The QLiberator compiler was used in the creation of the command package, and both the QLIB_EXT and QLIB_RUN files are utilised.

Memory

The total memory used is quoted as about 30 KB plus 50 bytes per file in the directory; that is, about 40 KB for a directory of 200 files. Free memory is displayed on an indicator at bottom right of the JDIR screen. A Quill _doc file gives five pages of instructions and notes, which are quite adequate to use the command, and give some useful advice.

One limitation, which will not bother many users, is that JDIR requires a multiple (dynamic) ramdisk facility to be available, such as that provided by the Trump and Gold Cards; the new command makes use of ram7_ for temporary files. You can use a static ramdisk, but that necessitates formatting ram7_ in advance, to a sufficient size for JDIR's temporary files. A TV screen will not display the full directory listing, so a monitor is a "must".

There is no need for a long description of how the "program" works. An illustration of the screen directory A look at the simple list of its advantages given below, should be sufficient to recommend the product:

Medium name, free/total sector counts, and total number of files, are displayed at the top of every screen page.

Files are listed in static pages - no scrolling.

Screens can be paged backwards and forwards with Ctrl-up/down.

Up to 76 file names, in 4 columns, per screen page.

Display Stays

The directory display remains on the screen after you exit JDIR, making file names available on-screen for subsequent typing-in.

When TK2 is active, selective directories can be displayed - eg flp2_ . ext will display only those file names having the suffix _ext.

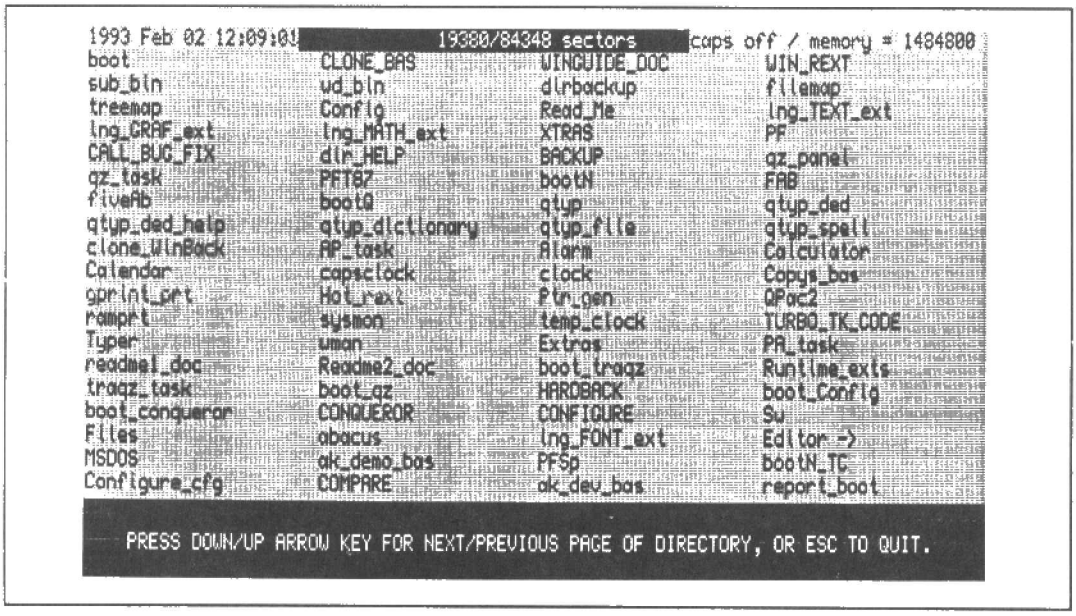
The print function gives up to 240 file names per A4 page (saving a great deal of paper, compared to doing screen dumps of the lists produced by the basic DIR command).

Output is selectable to screen, SER1, SER2, PAR, or device (mdv, flp, ram, NET etc.).

The JDIR command stays resident once loaded, until the QL is reset. It can be used from the SuperBasic command line, or in a program.

Free system memory is displayed on the directory screen.

Does it conflict with anything else? The only problem noted on the review system was hanging-up during the CALL phase of booting-up, with one version only, and this has been fixed. There has been a string of versions during the past few months, as (worthwhile) detail changes were made. One obvious feature lost along the way is numbering of the listed files; this had to go in order to get the current number of files on the display. Showing four columns of file names also meant limiting the name length, and the maximum displayed is 20 characters; anything beyond 20 is displayed as the first 19 plus a single full stop at the end of the name. While some improvement on DIR was made in the Trump/Gold Card rom, that was not a serious attempt at fixing the weaknesses; JDIR is much more comprehensive. The acid test is whether or not it will stay on one's boot disk for the next year or so, and my feeling is that the answer is "yes" in my case.



Of the areas in which the original QL design was clearly lacking, the DIR command is one that probably upsets almost all users. In its original form, it is singularly unhelpful: a fast-scrolling, single-column list of file names which has been and gone before you have time to note more than an odd name. Had some parameters been provided, to allow the file display to be modified, the command could have been made

"sensible" cost range - you have nothing much to lose.

Screen pages

The basic idea of the JDIR command is to deliver a list of files in screen pages, with four columns of up to 19 file names each. The supplied disk contains three separate sets of extensions files, loaded and made active by the usual RESPR, LBYTES and

QL-HDD-Card
Hard disk interface

Peta Jaeger packs everything onto one disk the only way - with a hard disk.

INFORMATION

Products: HDD-Card hard disk interface 259DM; hard disk controller 140 DM; complete 20-MB hard disk system 699 DM; complete 80-MB system 849 DM.

Supplier: Jurgen Falkenburg, Computer Technik, Thanweg 36, D-7539 Ersingen, Germany. Tel. (from UK) 010 49 7231 81058.

UK distributor: W N Richardson & Co. (EEC), 18-21 Misbourne House, Chiltern Hill, Chalfont St. Peter, SL9 9UE. Tel. 0753 888866.

Many of you have the problem that one boot-disk isn't enough if you're working a lot with *Easymenu, Quill, or Archive*. You usually need at least one disk for data. If you are working with Professional Publisher, you don't need just the boot-disk (which is full anyway) but also one or two disks for the data, one for publisher pages, one for fonts, Quill_docs, one for graphics, and so on. Since the QL is a multitasking system, you also get problems if you work with different programs (type into a word processor, switch to Archive for an address, use *Qpac2* for looking through your disks, and so on) within one session (and the program you need is always on the other disk!). Another annoying aspect of disks is their access time. If you load *Professional Publisher* from disk you must wait ... This is why I was always looking for a medium with more space than 720 KB.

The Gold Card came to my rescue. It is capable of accessing 3.2 MB disk drives. But very soon I found out that even 3.2MB isn't enough, since everything I needed, I copied to my boot-disk. It took a fairly long time to structure my large 3.2 MB disk just to cover a standard QL session. It often happened that I needed a program or data which were not on my boot-disk. There I go again, changing disks. The access time of the ED-disks is absolutely sufficient for QL programs. Loading or saving programs on 3.2 MB disks take no longer than 10 seconds (including Professional Publisher with all fonts). Quill and Archive load in less than a second.

Hard disks

Now let me tell you something about hard disks. For the folks who are not interested in the details, here is my opinion: go ahead and get one, it's worth the money.

If you want to connect the hard disk to your QL you need:

1. a QL
2. a hard disk interface such as the QL-HDD-Card from Jurgen Falkenberg
3. a memory card (Gold Card is best)
4. an adapter (to connect the interface with the QL)
5. a hard disk controller (Omti or Western Digital type for the HDD interface)
6. a compatible hard disk, as big as possible (MFM or RLL type in this case)
7. a power supply for the hard disk
8. a backup program

1 and 3 on this list are absolutely necessary. The interface is physically about 5 x 5 inches and cannot be push-fitted to the QL. Also, it does not have a through-connector (that's why you need number 4, the adapter). The interface is designed mainly for people who have put their QL into a bigger box (a PC housing, for example). That's why you get different adapters for your QL. Jurgen Falkenberg should have one which fits inside your housing. The QL-BUS-Expander is a good choice, since it is inexpensive and has two slots, so is suitable for use with the Gold Card. The BUS-Driver-Card is interesting if you do not have a Gold Card and want to use other expansion cards like the ROM-card, Eprommer, or any card which uses the expansion slot. If you are using the QL-BUS-Expander, just put your memory-card in the first and the interface in the second slot. There should be no problems.

Installation

Now for the tricky part. The hard disk must be connected to the controller which must be connected to the interface. The controller fits into the interface at a 90 degrees angle towards the interface. I had to buy the additional angle adapter, so that controller and interface face each other. Now all three cards are parallel. One drawback is that the interface is not the same size as the controller, so you can't screw them together. But the connectors fit tightly together, so it is not that bad. The hard disk is connected to the controller by two cables which you usually get together with the drive. The proper connection are very well described in the manual of the hard disk Interface.

You can use MFM- or RLL-type hard disks with suitable controllers. Hard disks need more power than floppy drives. They also need 12 VDC and can consume a lot of current. That's why you need number 5, the power supply. Modern hard disks need just 10 watts or less, so try to get the newest one you can lay your hands on. You should also look out for 3.5in drives, as they need less power, space and usually are quieter.

The whole process of installation isn't for the faint-hearted, but is easy for those who have already rebuilt their QL. If you can't handle this electronics stuff you can get a 'ready to go' hard disk (from Falkenburg or elsewhere) which comes complete with housing and power supply. After installation you can switch your QL on and use the hard disk! But first of all you should format the drive, and find a free address slot for the HDD-card which is compatible with the Gold Card or other memory-cards. Those who own a Gold Card have no choice but to switch to address 49152 (which is the rom port). This procedure is very well explained in the manual. But anyone who owns an old Gold Card (with the yellow PCB) must switch to address 835584 since address lines A18 and A19 are not grounded on older Gold Cards. To find out if the chosen address is correct or not, simply run this Basic program:

```
PRINT
PEEK_W(adr),PEEK_W(adr+2)
```

If this line gives back 19195 and 1, the address is already used. You will find a table with possible addresses for common memory-cards in the manual, which is 10 pages thick and A5 in format. All other technical details are explained there, so I won't discuss them here any further.

All Gold Card owners must also remove the 68008 main processor, otherwise the Gold Card crashes. This is no drawback since you don't need the 68008 with the Gold Card anyway. After the reset you'll find a copyright message and the size of your hard disk if the drive is formatted; otherwise you'll see "0 MB".

Formatting

The next step is formatting. You will need to know certain technical details about your drive, which should be in the instructions or handbook for whichever drive you have bought. If you enter them, they also serve as a formatting protection. You can't format the hard disk before you have entered the parameters with the WIN_FORMAT command. The parameters you must enter are interleave-factor, size of the clusters, sectors, heads, tracks, precompensation start cylinder, and reduced write-currency start cylinder. Usually you won't be able to establish the last two numbers but fortunately you can set them to zero.

The interleave factor is the sector offset and is an important part concerning the speed of the hard disk. With the Gold Card you can set the interleave to one, (if your controller can make it) so you can get a transfer rates of up to 500 KByte per second. Cluster size is the smallest area which a file occupies on the hard disk (in sectors of 512 bytes). Greater values save memory, but files use more space on the disk. Under 40 MB you should set cluster to four; if you have a bigger hard disk, to eight. It is important not to exceed 65536 clusters. The number of sectors is always 17 with MFM-Disks and 26 with RLL-Disks. Number of heads is the next value which you must enter. After that you must enter the number of tracks and the aforementioned precompensation and reduced write-currency start cylinder. Here comes an example for my Seagate MFM 40MB disk (with Gold Card):

```
WIN_FORMAT
1,4,17,5,976,300,0
```

The number of tracks is often given as the absolute number of tracks, for example 977 cylinders with my Seagate. WIN_FORMAT wants to know the last number from zero on, which is 976 in this case. Now you are ready to format the disk, just as you did with floppy disks before:

```
FORMAT win1_NAME
```

This will take some time (depending on the size of your hard disk). With four heads it'll take about one minute for each 150 tracks. After completed format the number of all formatted and good clusters will appear. The hard disk is now operating. After RESET you'll find the size of your hard disk just next to the copyright message (in my example I get '40MB'). If you press F1/F2 the system will try to boot from hard disk. If you put a BOOT-DISK in flp1 it will not boot from it until you put the following line in your hard disk boot file:

```
100 IF NOT
FTEST(\flp1_BOOT):LRUN
flp1_BOOT
```

(If you are using Toolkit 2.) Minerva users will find out, that because Minerva resets so fast the hard disk driver can't 'find' the hard disk, as the hard disk itself needs a few seconds to power up. This is even worse with Minerva MkII, where you can switch off the memory checking! What you have to do is to wait 2 to 3 seconds and press Alt-Ctrl-Shift-Tab for reboot. This is very tiresome if you want Minerva to reboot automatically while you are not in the same room. I hope this will be fixed in future drivers. (Version 3.07 and above will wait longer for slow hard disks to catch up, says Jurgen.)

Loading

Loading speed is very fast. There is no program on the QL which doesn't load within 3 seconds. Ninety-nine percent of them load within fractions of a second. 40 MB is sufficient for most users if you don't put a large MS-DOS partition on the disk. If you plan to do this, it is wiser to buy a bigger hard disk. The HDD-Card supports hard disks of every size, but disks over 40 MB are harder to get and very expensive.

If you have such a large disk, you must start using subdirectories extensively, but (as Bryan Davies often says) this is a real pain because Qdos allows only 36 characters per filename, and subdirectory names are part of the filenames. If you want to use meaningful filenames and subdirectory names, you are limited to about three levels of subdirectories. For example, if you want to create a subdirectory with the name WORDPROC (short for word processing) and three more in the next level for QUILL, TEXT91 and PERFECTION with extra subdirectories for TEXT and FONTS you must abbreviate, or put a lot of files within one directory. The last solution isn't something you should do, since you won't find your important files easily if you look for them in a directory with 100 files or more. Some examples:

```
MAKE_DIR WORDPROC_
MAKE_DIR
WORDPROC_QUILL_
MAKE_DIR
WORDPROC_PERFECT_
MAKE_DIR
WORDPROC_TEXT91_
MAKE_DIR
WORDPROC_QUILL_TEXTS_
MAKE_DIR
WORDPROC_PERFECT_TEXTS_
```

In the last example you can see that there are just 13 characters left for the filename of the texts, which are effectively just 9 characters, since you usually have a three character extension (such as _doc in QUILL). This problem will stay with us for a long time, since this isn't caused by the operating system or the level-2 filing hard disk driver, but of Tony's level 2 filing system. Let's hope there will be better level-3 drivers in the near future. There are all standard level-2 commands (as used by the Gold Card) within the driver software which I list here:

```
MAKE_DIR - create a
subdirectory
SET_FBKDT - set the backup-
Date
SET_FVERS - set the version
number
w=FMAKE_DIR - like
MAKE_DIR but returns an error
code
w=FBKDT - like SET_FBKDT
but returns an error code
w=FVERS - like SET_FVERS
but returns an error code
```

The exact syntax and function of the commands are explained in the manual. There is also a possibility of direct sector editing in the same way as there is with floppy disks, for example:

```
OPEN #channel,"win1_*D2D"
GET #channel\sectorno, a$
CLOSE #channel
```

The cost of the system is very important to most of QL users. The hard disk card (HDD-CARD) costs 259 DM (There are 2.38 DM to the pound sterling as we write. Check with your bank or the City pages of your daily paper for a current comparison), and you also need a suitable controller at 140 DM, and a hard disk. You can get second-hand hard disks quite cheap, since a lot of MS-DOS users switch to AT BUS disks. A 40-MB disk costs up to 300 DM. A complete system with a 20 MB hard disk, case and power supply costs 699 DM from Jurgen Falkenberg, 549 DM without case and power supply. A 80-MB Set may be supplied at 849 DM.

This system is primarily designed for QL users who have their QL built into a PC-housing, or are planning to do so (which is a speciality of Jurgen's). Falkenberg also supplies quite a nice mini-tower housing for the QL. For those who have a lot of data to handle, something like the HDD-Card is a must. The MS-DOS emulator *Conqueror* works very well with the hard disk and brings an extra speed increase, since a lot of DOS software depends on a fast hard disk. I couldn't find any severe problems with QL software which doesn't run from hard disk nor could I find any hardware problems. The combination of QL, HDD-Card, and Gold Card has been working fine for more than eight months with my system, though my QL still operates with the normal QL power supply. My opinion: Very good.

Falkenburg produces an information leaflet on the QL-HDD-Card and its supporting boards, which is informative but assumes a certain knowledge of hard disks. The Falkenburg interface, and the systems he supplies, are designed for use with Omti or Western Digital Winchester disk controllers, and MFM or RLL type hard disks. These are widely used standard types, but if you are hoping to set up a QL hard disk system using the QL-HDD-Card, check with Jurgen or Bill Richardson that any outside hard disk unit and/or controller you wish to purchase is compatible with their system, and also to make sure you order the correct interface and/or expansion boards.

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT U.K.
Tel: Bangor (0248) 354023

QL SOFTWARE

NB PLEASE NOTE THE MEANING OF CODES IN SQUARE BRACKETS BELOW:
[R] RAMDISK REQUIRED
[F] AVAILABLE ON FLOPPY DISK
[M] AVAILABLE ON MICRODRIVE
[128K/512K] MINIMUM MEMORY REQUIRED.
[PC] FOR IBM PC AND COMPATIBLES.
[TK2] TOOLKIT 2 REQUIRED.

CATALOGUE

THE CATALOGUE CONTAINS MORE DETAILS OF THE PROGRAMS. CALL FOR A COPY, OR ASK FOR A COPY WITH YOUR ORDER.

DEMONSTRATION VERSIONS

Demo versions are available for some of our programs. They cost £2.00 each and this cost is refundable against the full price of the full version, see lists below.

FLOPPY DISKS & LABELS

3.5" DSDD UNBRANDED DISKS	£0.40
3.5" DSHD FLOPPY DISKS	£0.70
MICRODRIVE CARTRIDGES	£2.50
100 DISK LABELS	£2.00
100 DISK LABELS ON PRINTER ROLL	£2.50
100 ADDRESS LABELS ON ROLL	£2.00
100 MICRODRIVE LABELS ON ROLL	£2.00

Please add £0.50 postage per roll of labels (max postage £2.50 if only ordering labels. Add £2.50 per order for floppy disks. No extra postage on microdrive cartridges.

FILE TRANSFER

DISCOVER £20.00
[F 256k] Copy files QDOS to PC disks and vice versa.
MULTI DISCOVER £30.00
[F 256k Enhanced versions, extra formats.
TEXTIDY £15.00
[F M 256k] Strip Quill files etc down to plain text.
OPD INTERCHANGE £15.00
IC/OPD file transfer utility. Unsuitable Gold Card.
QL-PC FILESERVER £24.50
[F 128k] QL/PC software for serial cable link, access PC's drives, printer, transfer files, etc.

LEISURE

SOLITAIRE £15.00
[F M 128k] A very addictive patience card game.
CRICKET SECRETARY £12.00
[F M 128k] For cricket buffs everywhere!
QUESTION MASTER £10.00
[F M 128k] For revision or entertainment.
QUIZ SETS each £5.00
FLEET TACTICAL COMMAND II QL £39.95
[F 512k] Naval strategy game, play on 2 networked QLs.
FLEET TACTICAL COMMAND II PC £49.95
[PC] Licenced PC version of this superb game.
COMBINED QL & PC VERSIONS £69.95
Package deal for both QL and PC versions together.
FTC2 DATA PRINT UTILITY £9.95
[F 128k] Print out data from FTCII described above.
THE FUGITIVE £9.95
[F 512k] Text adventure game.
COCKTAILS WAITER £10.00
[F 256k] Database of 100's alcoholic drinks recipes.
RECIPE SETS each £5.00

PROGRAMMING

S EDIT editor £20.00
[F 384k] DEMO VERSION £2.00
EASYPTR III £60.50
[F 256k] Pointer environment programming aid.
EASYPTR III BUDGET VERSION £35.00
DISA DISASSEMBLER £29.00
[F 256k] Interactive pointer driven disassembler.
BASIC REPORTER £10.00
[F M 128k] BASIC programming aid, list varables, etc.
BUDGET QLIBERATOR COMPILER £25.00
[F M 128k] BASIC compiler for unexpanded QL.
QLIBERATOR BASIC COMPILER £50.00
[F 256k] Excellent BASIC compiler for the QL.
QLOAD & QREF UTILITY £15.00
[F M 128k 128k] Fast loader and cross reference utility.
MEGA TOOLKIT on disk £25.00
on EPROM cartridge and disk £40.00
Comprehensive toolkit of over 200 BASIC extensions.

FILE HANDLING

LOCKSMITHE £14.95
[M 128k] Microdrive backup program, byte for byte!
4MATTER & LOCKSMITHE £23.50
[F M 384k] Complete software backup system.
TOOLCHEST £14.95
[M F 256k] Create customised microdrive doctor.
FILES 2 £12.00
[F M 128k] Simple to use pseudo file handling utility.
FILEMASTER £12.00
[F M 512k R] For bulk copying of files, & disk labelling.
THE GOPHER £12.00
[F M 128k] File finder, for disk, mdv, or hard disk.
[F M 128k] DEMO VERSION £2.00
WINBACK £25.00
[F 256k] Hard disk and ED disk backup utility.
[F256k] DEMO VERSION £2.00

DISPLAY SOFTWARE

BANTER £25.00
[F 512k] Banner printing program for Epson etc printers.
VISION MIXER 1 £10.00
[F 512k] Screen display system.
VISION MIXER PLUS £22.50
[F 384k] Enhanced, menu driven version.
PICTUREMASTER £15.00
[F 256k] Screen drawing utility for use with VMixer, etc.
PICTUREMASTER PLUS £20.00
[F 384k] Enhanced version, extra features.
UPGRADE TO "PM PLUS" £5.00

PRINTER DUMPS

SIDEWINDER PLUS £24.95
[F M 512k] Screen and DTP page printer dump utility.
Suitable for 9 and 24 pin printers.

GRAPHICS

THE PAINTER V4.04 £25.00
[F 512k] 100% m/code pointer driven graphics program.
THE CLIPART £12.00
[F 128k] 3 disks packed full of screen pictures.
PICTORIAL INDEX FOR THE CLIPART £4.00
New printed index for the Clipart, with screen dumps.
QRACTAL £20.00
[F 512k] Machine code fractals program.
QRACTAL SCREENS £2.00
[F 128k] Sample screen pictures made with QRacta.
IMAGE PROCESSOR 2 £15.00
[F 512k] Graphics and picture editing, etc utility.
[F 512k] DEMO VERSION £2.00
PD2 CLIPART £10.00
[F 128K 2 disks of clipart screens.
SCREEN SNATCHER £10.00
[F M 128k] Grab screen displays from other programs.
TEXT 'N' GRAPHIX £20.00
[F 256k] Print text files containing screen dumps.
[F 256k] DEMO VERSION £2.00
TRANS24 £10.00
[M 128k] Translate 9 pin graphics for 24 pin printer.
SCREEN COMPRESSION £10.00
[F M 128k] NEW! Save space by compressing pictures.

TEXT

QTYP 2 £29.95
[F 512k] Spelling checker, uses pointer environment.
BIBLE TEXT DISKS, PLAIN TEXT FORMAT £20.00
BIBLE TEXT DISKS, QUILL FILE FORMAT £20.00
[F 512k] Text of the King James Bible on disks.
SPELLBOUND £30.00
[F M 384k] Spelling checker, non-pointer environment.
SPELLBOUND SPECIAL EDITION £50.00
[F 512k] Improved version with larger dictionary, etc.
UPGRADE TO SPECIAL EDITION £30.00
Please return master disk/cartridge when upgrading.
QUICK POSTERS £10.00
[F M 128k] Text poster maker for use with Star printers.

DATABASES

ADDRESS BOOK & LABEL PRINTER. £15.00
[F M R 384k] As used by DJC, database + label printer.
[F M R 384k] DEMO VERSION £2.00
DATA DESIGN £50.00
[F 512k] Programmable pointer environment database.
FLASHBACK £25.00
[F M 256k] Fast database written in machine code.
FLASHBACK SPECIAL EDITION £40.00
[F 256k] Enhanced version of Flashback.
QL GENEALOGIST SECOND EDITION £30.00
[F 384k] Family trees and family history program.
UPGRADE TO SECOND EDITION £12.00
Please return master disk with upgrade order.
BUDGET 128K GENEALOGIST £12.00
[F M 128k] Cut down version for unexpanded machines.
DISK INDEXER £12.00
[F M 256k] Make a database of contents of your disks.
DBEASY £15.00
[F 512k] Database front end for Archive etc.
DBPROGS £15.00
[F 512k] A collection of Archive utilities and text files.

QL HARDWARE

MINI PROCESS CONTROLLER £59.95
Relay switched outputs, controlled via QL serial ports.
SOFTWARE TOOLKIT FOR MPC £9.95
[F 128k] Extensions to simplify control the MPC.
NETWORK PROVER £3.50
A small box which plugs between networked QLs.
QPOWER REGULATOR £24.95

OTHER QL SOFTWARE

QPAC1 £19.95
[F 512k] Pointer environment utilities.
QPAC2 £30.95
[F 512k] Files maintenance menu, hotkeys etc.
QTOP £29.50
[F 256k] Desktop system, files, job control, etc.
PRINTERMASTER £20.00
[F M 128k] Printer control utility.
HOME BUDGET £20.00
[F M 128k] Domestic bills and accounts program.
REMIND-ME £12.00
[F M 128k] Dates reminder program. Very easy to use.
REMIND-ME PLUS £20.00
[F M 128k] Enhanced version.
SCREEN ECONOMISER £10.00
[F M 128k] Screen saver blanker.
SCREEN DAZZLER £15.00
[F M 384k] Like screen economiser but instead of blanking out the screen activates moving objects or screen effects to save screen burning, like some PC programs!
SLOWGOLD £5.00
[F 128k] Slowdown routine for Gold Card and any QL.
TASKMASTER £35.00
[F M 384k] Task switching utility.
DISK LABELLER £10.00
[F 256k] Prints neat labels for your floppy disks.
THE CAT £5.00
[F M 128k] List filenames in columns.
ROB ROY BARGAIN PACK £10.00
[F M 128k] Reviewed in QL World August 1991.

THE SMALL PRINT! POSTAGE AND PACKING CHARGES: Software is sent postfree to UK addresses. To other countries, please add £1.00 per program for postage and packing (sent by airmail where possible). PRICES: All prices are shown in UK Pounds Sterling. PAYMENT: We can accept payment by cheque (in UK Pounds Sterling currency only, please) drawn on UK branch of a bank or building society, by Eurocheque with card number written on the back, Postal Order, or by these credit cards: VISA, ACCESS/MASTERCARD, EUROCARD or by CONNECT card. Please state the card type, number, expiry date, your address, and sign orders sent by post. We can also accept orders paid by credit card over the telephone. There is an answering machine for when I am unable to answer in person so that I can call you back later. Goods remain the property of DJC until paid for in full. PLEASE STATE IF YOU REQUIRE SOFTWARE ON 3.5 OR 5.25 INCH DISKS OR MICRODRIVE CARTRIDGE.